A New Model for Simultaneous Balancing and Cyclical Sequencing of Asynchronous Mixed-Model Assembly Lines with Parallel Stations

Thiago Cantos Lopes
Federal University of Technology - Paraná (UTFPR)
Graduate Program in Electrical and Computer Engineering (CPGEI)
Av. Sete de Setembro, 3165, Curitiba-PR, Brasil, 80230-901
thiagolopes@alunos.utfpr.edu.br

Celso Gustavo Stall Sikora
Federal University of Technology - Paraná (UTFPR)
Graduate Program in Electrical and Computer Engineering (CPGEI)
Av. Sete de Setembro, 3165, Curitiba-PR, Brasil, 80230-901
sikora@alunos.utfpr.edu.br

Adalberto Sato Michels
Federal University of Technology - Paraná (UTFPR)
Graduate Program in Electrical and Computer Engineering (CPGEI)
Av. Sete de Setembro, 3165, Curitiba-PR, Brasil, 80230-901
amichels@alunos.utfpr.edu.br

Leandro Magatão
Federal University of Technology - Paraná (UTFPR)
Graduate Program in Electrical and Computer Engineering (CPGEI)
Av. Sete de Setembro, 3165, Curitiba-PR, Brasil, 80230-901
magatao@utfpr.edu.br

ABSTRACT

Mixed-model assembly lines often operate cyclically and may have parallel stations to increase productivity. Simultaneously balancing tasks and sequencing products allows achieving higher steady-state efficiency (average production rate). A steady-state representation model has already been described for mixed-model balancing; however, this model does not allow neither parallel stations nor simultaneous balancing and sequencing. This paper extends such previous MILP model to allow parallel stations and to combine balancing and sequencing. A simplifying hypothesis on product entry and departure orders is incorporated to reduce the search-space. The model was applied to a literature dataset and provided answers better than the best previously reported for 33 of the 36 instances; the proposed model proved the optimal answer of 27 instances, while no proven optimal answer for this dataset was yet published. The average reduction in cycle time in comparison to the literature was 14% with a correspondent 16% increase in production rate.

KEYWORDS. Mixed-Model Assembly Line, Cyclical Steady-State Optimization, Parallel Stations

Paper Topics: AD&GP - OR in Administration & Production Management, IND - OR in Industry

1. Introduction

This paper presents a mixed-integer linear programming model (MILP) and case studies for *simultaneous balancing* and *cyclical sequencing* of *mixed-model asynchronous assembly lines* with *parallel stations*. Each of the concepts highlighted in italic are hereafter introduced, along with a literature review of related works.

Assembly lines are productive systems designed for mass-production of similar products [Scholl, 1999]; they are commonly present in multiple industries: automobile, electronics, white goods, appliances, etc.. On its simplest form, an assembly line consists of a set of workstations positioned in a serial manner so that the assignment of tasks to stations is product-oriented (flow-shop) rather than process-oriented (job-shop). Stations have a certain processing time defined by the sum of the durations of tasks assigned to them. The time between two consecutive products (cycle time) is dictated by the station with the highest processing time, therefore, it is necessary to balance the task distribution in order to have good production rates. The assembly line *balancing* problem consists on optimizing the assignment of tasks to stations in regard to a certain goal, in general efficiency maximization [Scholl e Becker, 2006].

The cycle time's definition is simple when the line is dedicated to a single product model. However, these productive systems can be used to produce multiple different products leading to difficulties in performance measures [Merengo et al., 1999]. If significant set-up times exist between models, they are usually separated in batches and the line is called a multi-model one [Boysen et al., 2007]. If set-up times are negligible, the assembly line is referred to as a *mixed-model* one [Battaïa e Dolgui, 2013]. In this case, to optimize the line also implies on a mixed-model *sequencing* problem [Boysen et al., 2009]. If demands are reliably stable, this system can be optimized by cyclical sequences, in this case a *cyclical sequencing* problem arises. This class of cyclical problems employ the minimal part set (*MPS*) concept [Levner et al., 2010]: If one needs to produce 500 units of product one and 100 units of product two, the *MPS* is five units of product one and one unit of product two. Cyclical schedules are defined based on the problem's *MPS*.

Assembly lines can be further classified in terms of line pace [Boysen et al., 2008]: If the conveyor belt moves all pieces constantly, the line is called paced or continuous. If all pieces move together, but discreetly when the processing at their current station is complete, the line is synchronous unpaced one. Finally, if each piece is allowed to move to the next station independently (provided processing is complete and the next station is free), the line is referred to as an *asynchronous* one. Naturally, these line's controls affect the scheduling of pieces in a mixed-model sequencing problem. Two common features of asynchronous lines are blockages and starvations: blockages occur when a piece has completed processing at its current station, but cannot move to the following one because it is occupied; starvations occur when a piece departs a workstation and the next piece has still not completed processing at the previous station, leaving the workstation temporarily empty.

Lusa [2008] provides a review of parallelism variants in assembly lines. It is possible to have parallel independent lines, which can either be dedicated to the same product models or to different ones, it is also possible to have different rates of the same products in each line. Parallel lines can have crossovers, which allow greater flexibility and connect the lines. When all stations present crossovers one can say the assembly line has *parallel stations* [Boysen et al., 2008]. Parallelism allows greater flexibility by allowing pieces to stay longer at stations without compromising the production rate.

It is common for authors to use decomposition and (meta-)heuristic procedures to solve the combined mixed-model balancing sequencing problem [Battini et al., 2009; Özcan et al., 2010; Tiacci, 2015]. However, there are some noteworthy exceptions: Sawik [2002] compared *simultaneous* to sequential balancing and (non-cyclical) sequencing of mixed-model assembly lines with a MILP model. He verified that the combination of the degrees of freedom can lead to better answers. Sawik [2012] presented a MILP model with makespan minimization formulation for cyclical lines. Öztürk et al. [2015] developed a Constraint Programming (CP) formulation for the same problem of Sawik [2012], along with a dataset with 36 different cases. These cases can serve as a testbed for such difficult problem that combines balancing and sequencing features into the same framework. Both Öztürk et al. [2015] and Sawik [2012] employ makespan minimization formulations that optimize a few replications of the *MPS*.

In a recent work Lopes et al. [2016] used a MILP model to demonstrate that makespan minimization formulations do not necessarily optimize steady-state, and presented an alternative steady-state formulation. This formulation is shown to generate better cyclical behavior than the makespan minimization one. Figure 1 illustrates the differences between optimal cyclical schedules generated by the makespan minimization formulation and by the steady-state representation one. Lopes et al. [2016] confirmed the connections between balancing and sequencing highlighted by Sawik [2002]. However, Lopes et al. [2016] consider cyclical sequencing as a problem's parameter and do not address parallelism. These limitations prevent the authors from applying said model to Öztürk et al. [2015]'s dataset, as the product sequences are not given and the majority of the dataset refers to instances with parallel stations. In this paper, a MILP model is provided to overcome said limitations: The steady-state representation provided by Lopes et al. [2016] is enhanced to incorporate parallelism and sequencing decision variables, allowing a comparison to Öztürk et al. [2015]'s results on the originally proposed dataset.

This paper is structured as follows: Section 2 presents the hypotheses used to describe the studied problem. Section 3 presents the developed MILP model with the extended steady-state representation formulation. Section 4 discusses the results of applying the presented model to Öztürk et al. [2015]'s dataset. Section 5 discusses the results and the relevance of a particular model hypothesis presented in Section 2. Section 6 summarizes the main conclusions drawn from this paper, and presents directions for further works.

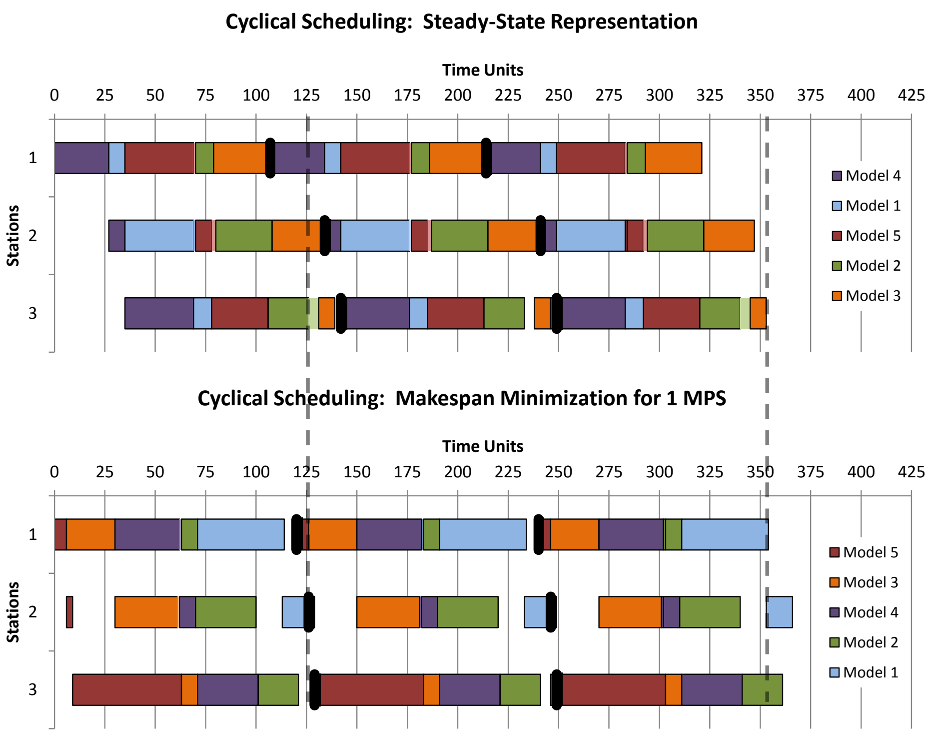


Figure 1: Comparison of solutions generated by each formulation. Black markers separate *MPS* replications, and dashed lines compare the solutions: Notice that the makespan minimization formulation outperforms the steady-state representation for the first *MPS*, but gradually loses that advantage in the steady-state.

2. Problem Statement

The simultaneous balancing and cyclical sequencing mixed-model asynchronous assembly line problem was described using a Mixed-Integer Linear Programming (MILP) mathematical model, presented hereafter. The proposed model is based on the following assumptions:

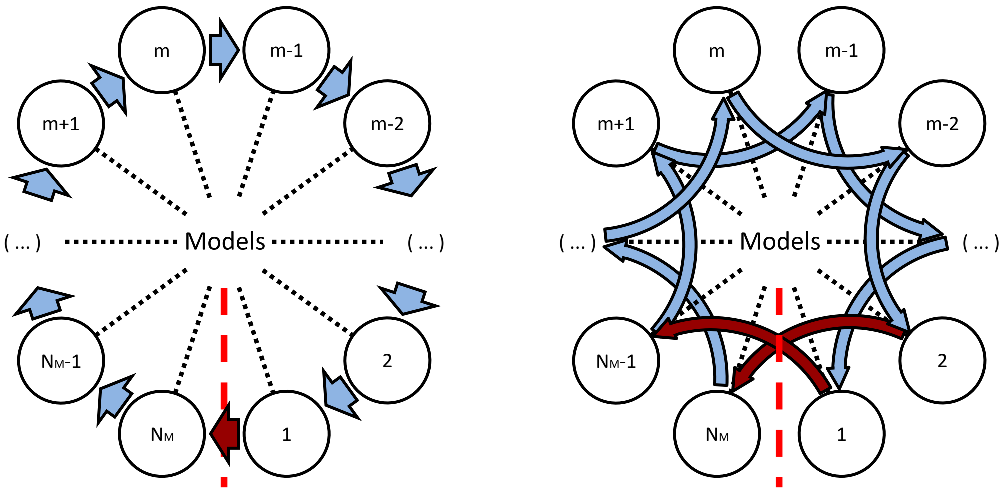
1. **Tasks and Assignments -** Tasks are indivisible and performed on the same stations for all product models, some tasks can only be assigned to a subset of stations;
2. **Precedence Restrictions -** Tasks must be assigned to either the same or following stations as their predecessors, as defined by the set of precedence restrictions;
3. **Line Pace -** The product flow is asynchronous and discrete. Pieces can only move to the next station when their processing is complete at their current station and the next one is empty;
4. **Cyclical Sequence -** The product sequence is cyclical and repeats indefinitely. The product sequence is not given and must also be optimized;
5. **Equal Parallelism Hypothesis -** All stations have the same parallelism degree k;
6. **"Instant Transportation" Hypothesis -** Transportation times between stations are negligible and can be ignored;
7. **Steady-State Optimization Goal -** The goal is to balance the line in a way that maximizes its efficiency, minimizing the average steady-state cycle time;
8. **Ordering Hypothesis -** The order products enter a station is the same order they depart it.

The first four assumptions state the base problem definition. The Equal Parallelism Hypothesis is employed to make analysis simpler, and also to fit the definition of a literature dataset [Öztürk et al., 2015]. The instant transportation hypothesis can be easily removed, but that would require additional parameters which are not defined by the literature dataset. In order to make consistent comparisons, therefore, such times are considered negligible. The goal consideration is to maximize steady-state efficiency (similarly to Öztürk et al. [2015]). However, the employed goal function mathematical definition is different.

The ordering hypothesis naturally holds in assembly lines with single stations, but it is a simplifying hypothesis for lines with parallel stations: Cyclical schedules do not necessarily impose that the entry and departure orders at each station be the same. However, this hypothesis simplifies modeling sequencing decisions and can also be seen as a search-space reduction heuristic that can aid convergence. Some insights on the relevance of this hypothesis are presented in Section 4 and Section 5.

The ordering hypothesis defines a concept for cyclical scheduling that is illustrated by Figure 2. In a line with single station, the second piece can enter the line when the first leaves, as shown by Figure 2a. In a line with parallelism degree of 2 ($k = 2$), the first and second piece enter independently, the third piece can enter when the first one leaves, and the forth piece can enter when the second one leaves (as shown by Figure 2b). The first and last piece define cycle time for lines single stations (as shown by Lopes et al. [2016]). Analogously, in lines with parallelism degree of two, both the first and second-last and the second and the last pieces constrain the value of average steady-state cycle time.

This concept can be naturally generalized for any degree of parallelism and any *MPS* size, even when there are more parallel stations than pieces in the *MPS*. The formulation alterations are presented for generality sake (section 3.2.1), but are not the focus of this paper.



(a) Single stations (k = 1) **(b)** Double stations (k = 2)

Figure 2: Cycle concept for lines with single and double stations: lighter arrows indicate constraints that describe the asynchronous piece flow, while darker ones indicate constraints that bind cycle time. This occurs because the later tie pieces in different *MPS* replications.

3. Mathematical Model

3.1. Nomenclature and Goal Function

The MILP model's nomenclature is defined by Table 1, which presents the parameter and variable sets of the problem, along with their domain and a brief description. The variable sets are presented along with the tuple sets for which they are defined, except for ct_{mix} which is a single real non-negative variable. In order to simplify notation understanding, lowercase letters are used for variables sets and uppercase ones are used for parameter sets. The developed model is a generalization of the one presented by Lopes et al. [2016] to which the capacity to describe parallel stations is added.

The problem's goal function is to minimize ct_{mix}, the steady-state cycle time of the full *MPS*, as stated by the Expression 1. This cycle time measures the time required for the full piece set to start repeating itself on the steady-state. It is determined by combining balancing, sequencing and scheduling aspects, as described in the Section 3.2.

$$\text{Minimize} \quad ct_{mix} \tag{1}$$

3.2. Constraints

The balancing aspect of the model is controlled by the binary variable set x: The Equation 2 states that all tasks must be performed at some station, and that only feasible task-station assignments are allowed (as defined by the set F). The Inequality 3 states the precedence relations (as defined by the set P) between tasks in terms of their station-wise assignments. The Equation 4 defines the processing time of each model at each station in terms of the tasks assigned to each station. At each station, the sum of processing times (divided by the parallelism degree) establish a lower bound for steady-state efficiency, as stated by the Inequality 5.

$$\sum_{(t,\, s)\, \in\, F} x_{\,[t,\, s]} = 1 \qquad \forall\, t \,\in\, T \tag{2}$$

$$\sum_{(t_1,\, s)\, \in\, F} s \cdot x_{\,[t_1,\, s]} \leq \sum_{(t_2,\, s)\, \in\, F} s \cdot x_{\,[t_2,\, s]} \qquad \forall\, (t_1,\, t_2) \,\in\, P \tag{3}$$

Table 1: Nomenclature of Parameters, and Variables: Domains and Descriptions

Parameters	Description
M	Set of models m from 1 to N_M, the number of models
S	Set of stations s from 1 to N_S, the number of stations
T	Set of tasks t from 1 to N_T, the number of tasks
F	Set of feasible tasks station allocations tuples $(t,\ s)$
P	Set of precedence relations tuples $(t_1,\ t_2)$
$D_{\ [T,M]}$	Durations of tasks for each model, measured in time units
k	Degree of parallelism

Variables	Description
$x_{\ [TS]}$	1 if task t is assigned to station s, 0 otherwise
$y_{\ [M,M]}$	1 if model m is the n^{th} piece in the sequence, 0 otherwise
$z_{\ [M,M]}$	1 if model m follows model n in the *MPS*, 0 otherwise
$zc_{\ [M,M]}$	1 if model m follows model n between *MPSs*, 0 otherwise
$t_{in\ [M,S]}$	Entry time of piece m in station s, measured in time units
$t_{x\ [M,S]}$	Processing time of model m in station s, measured in time units
$t_{out\ [M,S]}$	Departure time of model m in station s, measured in time units
ct_{mix}	Cycle time of the *MPS*, measured in time units

$$t_{x\ [m,\ s]} \ = \sum_{(t,\ s)\ \in\ F} D_{\ [t,\ m]} \cdot x_{\ [t,\ s]} \qquad \forall\, m \ \in \ M,\ s \ \in \ S \tag{4}$$

$$ct_{mix} \ \geq \ \frac{1}{k} \cdot \sum_{m\ \in\ M} t_{x\ [m,\ s]} \qquad \forall\, s \ \in \ S \tag{5}$$

The sequencing aspect of the model is mainly controlled by the binary variable sets y, z, and zc: The Equation 6 states that each position in the sequence is taken by a model. The Equation 7 states that each model takes a position in the sequence. The Inequality 8 states that the model m_2 "follows" the model m_1 if it takes the k^{th} position after model m_1's one. The Inequality 9 states that the model m_2 "follows" the model m_1 between *MPS* replications if it takes the $(N_M - k)^{th}$ position after model m_1's one. This concept is illustrated by Figure 2. Because the sequencing is cyclical by hypothesis, one can arbitrarily define the starting point and provide the model a symmetry break, for instance, by stating that model 1 is the first one in the sequence ($y_{\ [1,\ 1]} = 1$).

$$\sum_{m\ \in\ M} y_{\ [m,\ n]} \ = \ 1 \qquad \forall\, n \ \in \ M \tag{6}$$

$$\sum_{n\ \in\ M} y_{\ [m,\ n]} \ = \ 1 \qquad \forall\, m \ \in \ M \tag{7}$$

$$z_{\ [m_2,\ m_1]} \ \geq \ y_{\ [m_1,\ n]} + y_{\ [m_2,\ n+p]} - 1 \qquad \forall\, m_1, m_2, n \ \in \ M :\ n \ \leq \ N_M - p \tag{8}$$

$$zc_{\ [m_2,\ m_1]} \ \geq \ y_{\ [m_1,\ n]} + y_{\ [m_2,\ n+N_M-p]} - 1 \qquad \forall\, m_1, m_2, n \ \in \ M :\ n \ \leq \ p \tag{9}$$

Once balancing and sequencing are defined, the scheduling aspect of the model is controlled by the real variable sets t_{in} and t_{out}: The Inequality 10 states that a model can only leave a station after processing is finished. The Equation 11 states that when a model leaves a station it enters the next one. The Inequality 12 states that models cannot overlap in stations: if model m_2 "follows" the model m_1 (as controlled by the variable set F) then m_2 can only enter a station after m_1 has left it.

$$t_{out\ [m,\ s]} \ \geq \ t_{in\ [m,\ s]} + t_{x\ [m,\ s]} \qquad \forall\, m \ \in \ M,\ s \ \in \ S \tag{10}$$

$$t_{in\ [m,\ s]} \ = \ tout\ [m,\ s-1] \qquad \forall\, m \ \in\ M,\ s\ \in\ S:\ s>1 \tag{11}$$

$$t_{in\ [m_2,\ s]} \ \geq\ t_{out\ [m_1,\ s]} - BigM \cdot z_{\ [m_2,\ m_1]} \qquad \forall\, m_1, m_2 \ \in\ M,\ s\ \in\ S \tag{12}$$

The last model constraint is the one that ties steady-state cycle time to the scheduling variables: The Inequality 13 states that models cannot overlap in stations between two minimal part sets (*MPS*). This inequality is valid when the parallelism degree k is smaller or equal than the number of pieces in the *MPS* (N_M), the most common practical condition. The variable ct_{mix} states the time between entries of two equal pieces on the bottleneck station. Notice that this implies that the cycle time is the same for all pieces.

$$ct_{mix} + t_{in\ [m_2,\ s]} \ \geq\ t_{out\ [m_1,\ s]} - BigM \cdot zc_{\ [m_2,\ m_1]} \qquad \forall\, m_1, m_2 \ \in\ M,\ s\ \in\ S \tag{13}$$

3.2.1. Additional Constraints

The formulation presented thus far reflects expected practical conditions in which the parallelism degree k is smaller or equal than the number of pieces in the *MPS* (N_M). This was the case in all instances of Öztürk et al. [2015]'s data set, and is therefore the focus of this paper. In order to provide a more complete formulation for hypothetical cases in which $k > N_M$, the relevant model alterations are hereafter presented. First, to simplify notation, for each position j, a parameter $next(j)$ is defined in accordance to the Equation 14. This parameter represents positions that are connected according to the ordering hypothesis, i.e. the $next(j)^{th}$ piece will enter the station s when the j^{th} piece leaves it.

$$next(j) \ = \ (j+k-1) \bmod N_M + 1 \tag{14}$$

A modified version of the z and zc variables is presented. The binary variable $w_{\ [m_1,\ m_2,\ j]}$ is set to 1 when the model m_1 is assigned to the j^{th} position and the model m_2 to the $next(j)^{th}$ one, as stated by the Inequality 15.

$$w_{\ [m_1,\ m_2,\ j]} \ \geq\ y_{\ [m_1,\ j]} + y_{\ [m_2,\ next(j)]} - 1 \qquad \forall\, m_1, m_2, j \ \in\ M \tag{15}$$

The Inequality 16 replaces both Inequality 8 and Inequality 9. It ties entrance and departure times to the production mix cycle time in case $k > N_M$. This constraint was not employed during testing as none of the instances displayed that property. It is only provided here for generality sake.

$$ct_{mix} \cdot \left\lceil \frac{next(j)+k}{N_M} - 1 \right\rceil \ + \ t_{in\ [m_2,\ s]} \ \geq\ t_{out\ [m_1,\ s]} \ - \ BigM \cdot w_{\ [m_2,\ m_1,\ j]} \qquad \forall\, m_1, m_2, j \ \in\ M,\ s\ \in\ S \tag{16}$$

4. Results

In order to verify how well the proposed model performs on combinatorial cases, it was applied to the dataset presented by Öztürk et al. [2015]. Minor adaptations were required to incorporate the features of said dataset, for instance, tasks were subject to space constraints at stations. All the computational tests of this section were conducted on a 64 bit Intel™ i7 CPU (2.9 GHz) with 8 GB of RAM, using eight threads and the IBM ILOG CPLEX Optimization Studio 12.6 solver. Öztürk et al.'s model and instances are available as Supporting Information of their article. Here we used the script for model execution as it was made available by said authors. Thus, a comparison with the results obtained by the article's methodology, at the same hardware conditions, was possible. Öztürk et al. define different values of cycle times for models: they are measured by the time between the departures of two pieces of the same model from the last station. The proposed model allows one single value of cycle time for the whole *MPS*, the steady-state value.

Table 2: Comparison of results from Öztürk et al. [2015] and the proposed model. The column Problem Data contains the parameters of each of the 36 instances. WC, AV, UB, LB, TBS, and RT stand for: the Worst Case (WC) and the Average Value (AV) for the Öztürk et al.'s model, the Upper Bound (UB) and Lower Bound (LB) from the proposed model, the necessary Time to reach the Best Solution (TBS) obtained by each approach and the total Run Time (RT) in seconds, respectively.

Problem Data					Öztürk et al. [2015]				Proposed Model				Comparisons	
n	N_T	N_M	N_S	k	WC	AV	TBS [s]	RT [s]	UB	LB	TBS [s]	RT [s]	WC/UB	AV/UB
1	38	5	3	1	47	47	2354	3600	47	47	0.2	0.2	1.00	1.00
2	38	5	3	2	26	24.8	1304	3600	23.5	23.5	0.1	0.1	1.11	1.06
3	38	5	3	3	19	17	1130	3600	15.7	15.7	0.1	0.1	1.21	1.09
4	38	5	5	1	26	26	599	3600	26	26	0.4	0.4	1.00	1.00
5	38	5	5	2	14	13.6	859	3600	13	13	0.3	0.3	1.08	1.05
6	38	5	5	3	12	10.6	62	3600	8.7	8.7	0.3	0.3	1.38	1.22
7	52	7	3	1	59	59	402	3600	57	57	6.1	9.2	1.04	1.04
8	52	7	3	2	32	30.4	375	3600	28.5	28.5	5.5	8.0	1.12	1.07
9	52	7	3	3	25	24.1	105	3600	19	19	5.9	10.2	1.32	1.27
10	52	7	5	1	34	34	1588	3600	34	34	1.4	1.6	1.00	1.00
11	52	7	5	2	18	18	672	3600	17	17	0.7	1.1	1.06	1.06
12	52	7	5	3	14	14	230	3600	11.3	11.3	1.5	1.9	1.24	1.24
13	114	5	3	1	117	117	231	3600	107	107	6.4	7.3	1.09	1.09
14	114	5	3	2	67	65.8	921	3600	53.5	53.5	7.2	7.8	1.25	1.23
15	114	5	3	3	50	41.6	3091	3600	35.7	35.7	6.1	6.5	1.40	1.17
16	114	5	5	1	72	72	1093	3600	68	68	53.5	54.6	1.06	1.06
17	114	5	5	2	39	37	534	3600	34	34	38.3	39.5	1.15	1.09
18	114	5	5	3	30	25.2	2536	3600	22.7	22.7	49.3	57.1	1.32	1.11
19	156	7	3	1	156	156	3430	3600	142	142	22.8	39.6	1.10	1.10
20	156	7	3	2	76	76	3592	3600	71	71	49.1	49.7	1.07	1.07
21	156	7	3	3	64	63.6	785	3600	47.3	47.3	50.5	52.3	1.35	1.34
22	156	7	5	1	93	93	1695	3600	88	88	672	1352	1.06	1.06
23	156	7	5	2	65	64.4	3295	3600	44	44	1528	2127	1.48	1.46
24	156	7	5	3	36	34.3	3510	3600	29.3	29.3	2302	3449	1.23	1.17
25	190	5	3	1	197	197	1529	3600	197	197	0.2	0.3	1.00	1.00
26	190	5	3	2	112	108.4	1560	3600	98.5	98.5	0.1	0.2	1.14	1.10
27	190	5	3	3	81	72.4	1160	3600	65.7	65.7	0.1	0.3	1.23	1.10
28	190	5	5	1	124	124	2271	3600	113	113	198	561	1.10	1.10
29	190	5	5	2	73	65	2509	3600	56.5	56.5	1360	1679	1.29	1.15
30	190	5	5	3	52	51.8	2017	3600	37.7	37.7	1120	1125	1.38	1.38
31	260	7	3	1	267	267	2620	3600	263	263	2.0	6.4	1.02	1.02
32	260	7	3	2	149	147.4	2722	3600	131.5	131.5	3.9	4.3	1.13	1.12
33	260	7	3	3	132	128.7	3158	3600	87.7	87.7	7.1	7.5	1.51	1.47
34	260	7	5	1	240	236	660	3600	155	129.1	1078	3600	1.55	1.52
35	260	7	5	2	98	97.1	2843	3600	77.5	64.8	743	3600	1.26	1.25
36	260	7	5	3	81	78.6	2784	3600	52	43.3	1311	3600	1.56	1.51
		Average			76	78	1673	3600	66	65	295	596	1.20	1.16
		Maximum			267	267	3592	3600	263	263	2302	3600	1.56	1.52

The Table 2 presents the obtained results for both the proposed formulation and Öztürk et al.'s formulation. WC presents the highest value of cycle time amongst models found by Öztürk et al.'s formulation. AV presents the average value of cycle time obtained when all models are considered. UB and LB present the upper (primal) and lower (dual) bounds for the best answer the proposed model found for each case n. The necessary Time to reach the Best Solution (TBS) and the total Run Time (RT) are showed in seconds. For all cases, the time limit is set in 3600 seconds. The "Comparisons" column divides the best and average value found by our formulation by those obtained by the benchmark's author.

5. Discussion

The comparisons show that the proposed model outperformed Öztürk et al.'s one in 33 out of the 36 instances (on three of them, both formulations tied). In average, the proposed formulation allowed answers with 14% lower values of cycle time or 16% higher throughput. The average processing time for the proposed model was 596 seconds, which was primarily dictated by the eight cases that took more than 1000 seconds. However, the proposed formulation proved the optimality of 27 out of the 36 instances in less than a minute. Öztürk et al.'s formulation, on

Table 3: Illustrative problem data: processing times of each model at each station

Model	1	2	3	4	5
Station 1	12	4	4	4	6
Station 2	4	6	6	7	6
Station 3	6	6	4	7	4

the other hand, did not prove any optimal solution. The three instances whose answers are equal were obtained with a significant lower processing time with the proposed model (Case 1: 0.2 to 2354 seconds; Case 4: 0.4 to 599 seconds; Case 25: 0.2 to 1529 seconds).

The factors that seem to justify these differences are stated as follow: Contrary to Öztürk et al., the proposed formulation does not require multiple replications of the *MPS* to be taken into account. Furthermore, the proposed model is a MILP one instead of a CP (Constraint Programming) one and, therefore, allows scheduling variables to assume both integer and fractional values. Lastly, the ordering hypothesis (each product entry order must be the same as its departure order) also restricts the search field.

The ordering hypothesis is also relevant for the optimality of solutions: while in assembly lines without parallelism, the product order cannot be altered, in lines with parallel stations, this might be the case. This means that out of the 33 cases solved to optimality with the ordering hypothesis, only 11 of them have steady-state optimality assured for the general case (the eleven cases with $k = 1$ that were solved to optimality). The ordering hypothesis is not expressed as model's constraints. The proposed model was constructed based on the ordering of tasks' hypothesis. Therefore, in order to verify whether better solutions are possible or not, a new mathematical model and further tests would be required. A single example can, however, illustrate that relaxing the ordering hypothesis allows greater liberties, and potentially, better solutions.

In order to illustrate the additional liberties associated with the relaxation of the ordering hypothesis, a very small scale instance of the problem is defined according to the data presented by the Table 3. The processing times are known (balancing solution is given) and one is only required to sequence and cyclically schedule it in an assembly line with parallelism degree k equal to 2.

The optimal solution of the illustrative problem, according to presented model (i.e. with the ordering hypothesis) is presented by the Figure 3. The thick black markers separate the *MPS* replications. Notice that stations actually process the same pieces in an alternated manner. The solution generated by the model is equivalent to independent parallel lines, as each "side" of the line can function independently: Stations 1.1, 2.1 and 3.1 represent one "side", while Stations 1.2, 2.2, and 3.2 represent the other.

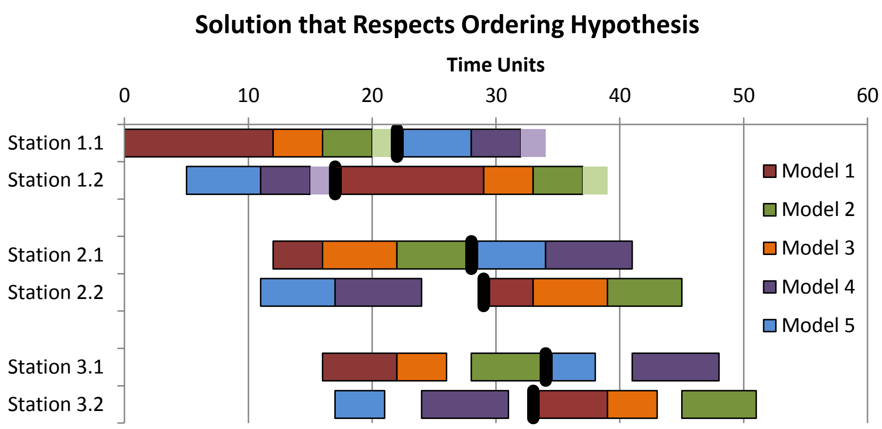


Figure 3: Solution with $ct_{mix} = 17$. All models are produced in both lines in an alternated manner.

It is also possible for independent parallel assembly lines to display a different behavior: One line can be used to produce one product mix and the other for another product mix. This is illustrated by the Figure 4. This kind of answer respects the ordering hypothesis, but would require constraint alterations[1]. Solutions that respect the ordering hypothesis can, therefore, display either one behavior or the other, but it is not possible for it to produce solutions that combine these behaviors: to have some stations have one product sequence and other stations have a different product sequence. The ordering hypothesis demands that the cyclical sequencing is the same for all stations and, therefore, either all stations produce all products alternately (Figure 3) or all stations produce different product mixes (Figure 4). In this later case, it is possible to separate them in groups or "sides" that process the same product mixes as illustrated by the Figure 4.

Figure 4: Solution with $ct_{mix} = 18$. Models 1 and 4 are produced in one side of the line and the others in the other side.

Parallel stations can, however, display a more flexible behavior, which is illustrated by the Figure 5: In the first and third stations, all products are processed in both "sides" of the assembly line. However, the cyclical product sequence in Stations 1.1 and 1.2 is different than in Stations 3.1 and 3.2: In the first set of stations, the model sequence is (1,2,3,4,5) while in the third one the sequence is (2,1,5,3,4)[2]. Furthermore, each of the Stations 2.1 and 2.2 processes a different set of products: Station 2.1 processes the product models 2 and 4, and Station 2.2 processes the product models 1, 3, and 5. In that sense, the solution in Figure 5 combines the solutions from Figures 3 and 4 by adding greater ordering flexibility.

The solution presented by Figure 5 outperforms the one presented by Figure 3, as it has a smaller value of ct_{mix}. This is only possible due to the greater flexibility offered by the removal of the ordering hypothesis: steady-state blockages and starvations can be reduced by allowing the cyclical product sequence to be different at each station. However, these differences must allow consistent cyclical schedules: it must be possible cyclically change the positions via scheduling by exploiting the cross-overs, as illustrated for the example in Figure 5.

This means that, if in one hand the ordering hypothesis and the cyclical concept presented by Figure 2 allowed the developed model to achieve better solutions than Öztürk et al. [2015] in 33 out of 36 instances, on the other hand, it is possible that better solutions exist in all 24 instances with parallelism: If a better cyclical schedule could be found for the illustrative case (in which balancing is given), it is likely that this extra degree of freedom would have positive syn-

[1]Instead of using the cyclical framework illustrated by Figure 2, have the piece order be (1,2,3,4,5) and have the logical ties occur between the pieces in the positions 1-5, 2-3, 3-4, 4-2, and 5-1. The piece order is the same at all stations, but the cyclical concept in Figure 2 is not compatible.

[2]In Figure 5, station 1.1 displays a (1,2,3,4,5) sequence, and Station 1.2 displays a (2,3,4,5,1), which is the same cyclical pattern. Analogously, station 3.1 displays (2,1,5,3,4) and station 3.2 displays (3,4,2,1,5), these are also the same cyclical patterns.

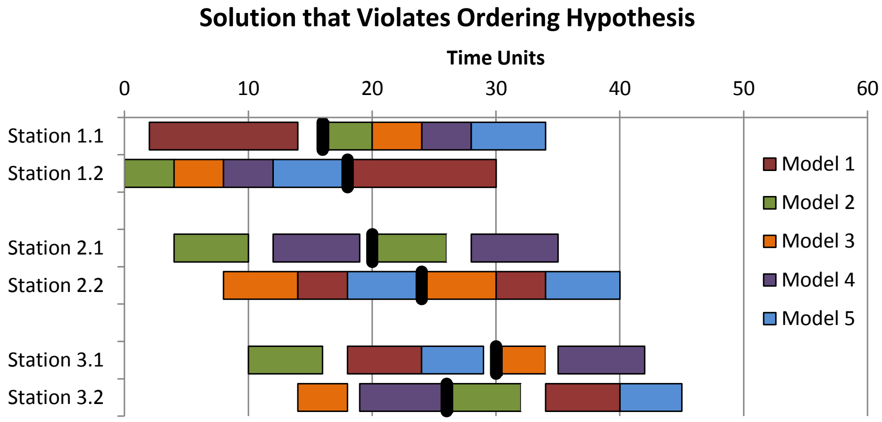


Figure 5: Solution with $ct_{mix} = 16$. In the first and third stations, all models are produced in an alternated manner, but in different orders. In the second station, different sets of products are produced in each side of the line.

ergies when combined to the balancing degree of freedom of the composed problem. This means that out the 33 optimal solutions found by the model (Table 2), only 11 (the ones with $k = 1$) are truly guaranteed to have optimal cyclical steady-state behavior. The others are only optimal in regard to the ordering hypothesis. The development of a further generalized model that can optimize the steady-state representation described by Lopes et al. [2016] and herein enhanced for parallel stations, but without the ordering hypothesis is, therefore, a promising (yet challenging) modeling direction for further works.

6. Conclusions

This paper presents a new model for simultaneously balancing and cyclically sequencing asynchronous lines with parallel stations. The model generalizes a previously validated steady-state representation [Lopes et al., 2016] that requires only one replication of the *MPS*. A search-space reduction hypothesis (the ordering hypothesis) is incorporated in the proposed model, stating that the order pieces enter sets of parallel stations is the same order in which they leave them. This constraint simplifies modeling and allows a single binary variable set to represent sequencing decisions.

The proposed model is applied to a literature dataset, leading to results significantly better than those previously reported: solutions with smaller cycle time that the best previously reported ones were provided for 33 out of the 36 cases, with an average of 14% reduction in cycle time or an equivalent to a 16% increase in production rate (Table 2). Reasons behind these better results lie in a combination of factors: the proposed model is a MILP one that allows fractional variables, requires only one replication of the *MPS* to function, and incorporates a search-space reduction offered by the ordering hypothesis. The reference model [Öztürk et al., 2015], which provided the best formerly known answers, is a CP (Constraint Programming) one that does not allow fractional variables, requires multiple replications of the *MPS* and does not incorporate the equivalent space-reduction constraint.

The main model simplifying hypothesis is further exploited in a small problem instance that can be manually solved. This example is able to demonstrate that violations of ordering hypothesis can lead to better solutions: Parallel lines can allow cyclical steady-state schedules that have different cyclical product sequences in each stage. Solutions that respect the ordering hypothesis can either have all products be processed at all stations (Figure 3) or have each side of parallel stations only processing subsets of the produced models (Figure 4). By relaxing that hypothesis, it is possible to have parts of the solution with one behavior (stations that process all product pieces) while other parts display another behavior (division of the product pieces in a subset), as indicated in Figure 5. This means that while the ordering hypothesis helped to provide significantly better solutions on the tested literature dataset, it also might eliminate even better solutions on instances with parallel stations.

Therefore, further works should seek to develop a model that further generalizes the steady-state representation, by removing the ordering hypothesis. This is likely to lead to both modeling and computational challenges, as even with the search-space reduction hypothesis some cases where computationally intractable. In that sense, the ordering hypothesis can be seen as a simplifying heuristic or as a basis for further less restrictive search-space reduction constraints.

Acknowledgements

The authors would like to thank financial support from Fundação Araucária (Agreement 141/2015 and 06/2016 FA – UTFPR – RENAULT) and CNPq (grant 406507/2016-3).

References

Battaïa, O. e Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142:259–277.

Battini, D., Faccio, M., Persona, A., e Sgarbossa, F. (2009). Balancing - sequencing procedure for a mixed model assembly system in case of finite buffer capacity. *International Journal of Advanced Manufacturing Technology*, 44:345–359.

Boysen, N., Fliedner, M., e Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, 183:674–693.

Boysen, N., Fliedner, M., e Scholl, A. (2008). Assembly line balancing: Which model to use when? *Intern. Journal of Production Economics*, 111:509–528.

Boysen, N., Fliedner, M., e Scholl, A. (2009). Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research*, 192:349–373.

Levner, E., Kats, V., Alcaide, D., Pablo, L. D., e Cheng, T. C. E. (2010). Complexity of cyclic scheduling problems: A state-of-the-art survey. *Computers & Industrial Engineering*, 59(2):352–361.

Lopes, T. C., Sikora, C. G. S., e Magatão, L. (2016). Buffer and Cyclical Product Sequence Aware Assembly Line Balancing Problem: Model and Steady-State Balancing Case Study. In *Annals of the XLVIII SBPO*, p. 3458–3469, Vitória-ES, Brazil.

Lusa, A. (2008). A survey of the literature on the multiple or parallel assembly line balancing problem. *European Journal of Industrial Engineering*, 2(1):50–72.

Merengo, C., Nava, F., e Pozzetti, A. (1999). Balancing and sequencing manual mixed-model assembly lines. *International Journal of Production Research*, 37(12):2835–2860.

Özcan, U., Çerçioglu, H., Gökçen, H., e Toklu, B. (2010). Balancing and sequencing of parallel mixed-model assembly lines. *International Journal of Production Research*, 48(17):5089–5113.

Öztürk, C., Tunal, S., Hnich, B., e Örnek, A. (2015). Cyclic scheduling of flexible mixed model assembly lines with parallel stations. *Journal of Manufacturing Systems*, 36:147–158.

Sawik, T. (2002). Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line. *European Journal of Operational Research*, 143:115–124.

Sawik, T. (2012). Batch versus cyclic scheduling of flexible flow shops by mixed-integer programming. *International Journal of Production Research*, 50(18):5017–5034.

Scholl, A. (1999). *Balancing and sequencing assembly lines*. Physica, Heidelberg, 2nd edition.

Scholl, A. e Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168:666–693.

Tiacci, L. (2015). Simultaneous balancing and buffer allocation decisions for the design of mixed-model assembly lines with parallel workstations and stochastic task times. *International Journal of Production Economics*, 162:201–215.