Traveling worker assembly line (re)balancing problem: model, reduction techniques, and real case studies

Celso Gustavo Stall Sikora, Thiago Cantos Lopes, Leandro Magatão*

Graduate Program in Electrical and Computer Engineering (CPGEI)

Federal University of Technology - Paraná (UTFPR), Curitiba, Brazil, 80230-901

Abstract

The assembly line balancing problem arises from equally dividing the workload among all workstations. Several solution methods explore different variants of the problem, but no model includes all characteristics real assembly lines might contain. This paper presents a mixed integer linear programming model that solves the Traveling Worker Assembly Line Balancing Problem (TWALBP). In this problem, the tasks' balancing along with the assignment of workers to one or more workstations is determined for a given layout. The assignment flexibility is solved with a traveling salesman problem formulation integrated in the balancing model. Adapted standard datasets and three real case scenarios are used as benchmark sets. These scenarios present particularities such as human and robotic workers, assignment restrictions, zoning constraints, automatic and common tasks. The model successfully determines the tasks' assignments and the routing of every worker for a layout aware optimization of assembly lines. Better quality balancing solutions were achieved allowing workers to perform tasks at multiple stations, showing a trade-off between assignment flexibility and movement time.

Keywords: Combinatorial optimization, Assembly line rebalancing, Real-world application, Traveling salesman problem, Mixed integer linear programming

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

European Journal of Operational Research, Volume 259, June 2017, Issue 3, Pages 949-971

DOI: 10.1016/j.ejor.2016.11.027

1. Introduction

The assembly line balancing problem (ALBP) tackles the problem of task allocation, deciding to which workstation each task should be assigned in regard to the problem's constrains. Common constrains for ALBP are precedence relations and assignment restrictions. This class of balancing problems might also have various goal functions such as minimizing the number of workstations,

*Corresponding author

Email address: magatao@utfpr.edu.br (Leandro Magatão)

given a cycle time (variant 1 or ALBP-1) and minimizing cycle time given a number of workstations (variant 2 or ALBP-2) (Scholl, 1999).

The simpler version of this problem (SALBP, Simple Assembly Line Balancing Problem) was first defined by Baybars (1986). Its main assumptions are: each task can be assigned in any workstation, the line produces only one homogeneous product, stations are equally equipped in respect to machinery and workers, the line is considered serial, no parallel stations or feeder lines exist, there is only one workplace per station, etc.

Optimization processes seek to minimize the production costs of the assembly line. During such processes, balancing is commonly defined in terms of workstations, a usual assumed hidden hypothesis in ALBP is that every workstation is equivalent to a worker. However, there are variations of the problem for which the station-worker disassociation is necessary. One example of such variation is when more than one worker per station is allowed (Fattahi et al., 2011; Yazgan et al., 2011). In this case, the assignment of tasks and workers must also consider interferences within the workstation (Boysen et al., 2008). Assembly lines with stochastic task times or mixed-models may also assign facility workers to deal with temporary unbalances of the system (Altemeier et al., 2010; Battaïa et al., 2015; Gronalt & Hartl, 2003; Gujjula & Gunther, 2009; Mayrhofer et al., 2013). Furthermore, a series of recent papers (Borba & Ritt, 2014; Costa & Miralles, 2009; Moreira et al., 2015; Sungur & Yavuz, 2015; Vilà & Pereira, 2014) treat the assignment of heterogeneous workers in workstations along with the balancing. The Assembly Line Worker Assignment and Balancing Problem (ALWABP), defined by Miralles et al. (2008), allows including disabled people in assembly lines. In ALWABP, the objective is to determinate both the allocation of tasks to stations and a single station for each worker.

If we consider that workers can be assigned to more than one station, and, therefore, have to move between stations, one particular variation of the ALBP arises: The Traveling Worker Assembly Line Balancing Problem (TWALBP), combining balancing features to a TSP formulation. The key difference in this model is that workers can move between stations to perform tasks, and each worker limits the cycle time by the sum of the movement and processing times of the tasks assigned to him/her. One significant advantage of this feature in comparison to fixed worker allocation is that, even though precedence constraints must hold for stations, they can be relaxed for workers. Workers are able to move between workstations, allowing them to perform tasks from "different regions" of the precedence diagram, while considering the movement time. An example instance for the herein defined TWALBP can be seen in Figure 1: a balancing with 6 workers is only achievable when we allow workers to move between stations.

A similar relaxation is presented in U-shaped lines. As defined by Miltenburg & Wijngaard (1994), in U-Shaped Line Balancing Problems, the distances between each side of the line is small. Therefore, workers can perform tasks in both the beginning and in the end of the precedence diagram, when either all predecessors or all successors are completed. By this reasoning, more assignment options are possible, usually resulting in a better quality balancing. This benefit is not restricted to U-lines: in theory multiple allocations for workers could increase the line efficiency

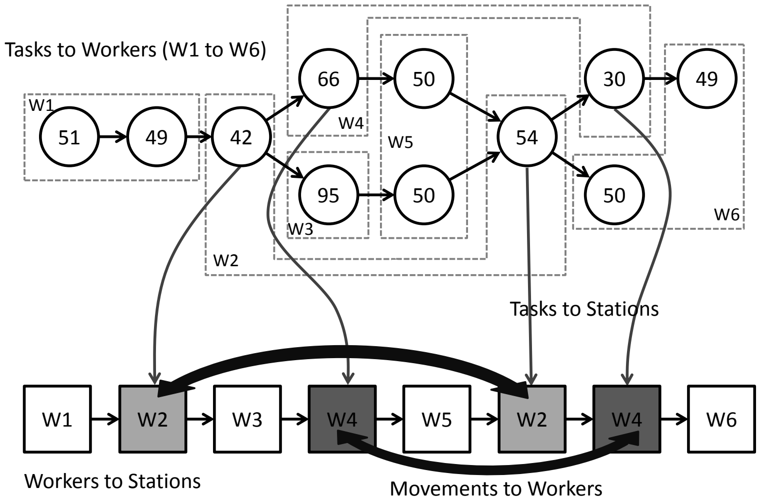


Figure 1: Example of a TWALBP instance for a cycle time of 100 time units. The time of each task is given inside its precedence diagram node. Notice how workers 2 and 4 are assigned to two stations each. This allows them to perform tasks at "different regions" of the precedence diagram. This answer is valid if the movement time of workers 2 and 4 are equal or lower than 4 time units. The dashed polyhedral shows the assignment of tasks to workers. The gray arrows illustrate the assignment of tasks to stations. The numbered squares show the worker's assignment while the movement between stations are illustrated by the thick black arrows. This TWALBP instance requires 6 workers and 8 stations. Their SALBP and ULBP versions would require 7 workers for the same cycle time.

for different line form.

Although the U-Shaped Line Balancing Problem (ULBP) was defined without considering movement times (Miltenburg & Wijngaard, 1994), subsequent papers aggregate them in their solution methods. Sparling & Miltenburg (1998) presented a non-linear formulation for the ULBP with movement times, but their method only calculated and added the dislocation time for a given solution of the task distributions. Sparling & Miltenburg's model considered a continuous assembly line, the workers moved along with the conveyor belt. Further researchers solved this formulation with different methods: dynamic programming (Miltenburg, 2001), heuristics (Shewchuk, 2008), and metaheuristics (Sirovetnukul & Chutima, 2010; Zha & Yu, 2014). Nakade & Ohno (2003) and Nakade & Nishiwaki (2008) considered walking times in the scheduling of workers in U-lines. Their formulations, however, considered the task assignment as given.

For the TWALBP, we consider that workstations have fixed positions and the movement time between any pair of stations is given. Within this assumption, the assignment of workers to stations can be seen as a Traveling Salesman Problem (TSP) and be modeled using Mixed-Integer Linear Programming.

Further practical considerations for ALBP are due to re-balancing scenarios. According to Falkenauer (2005), the reallocation of tasks of an operating line has several restrictions due to the difficulty of changing previously implemented decisions. Boysen et al. (2007), in their survey, pointed that re-balancing is usually modeled either by assignment restricted models (Bautista & Pereira, 2007; Scholl et al., 2010; Sternatz, 2014) or by cost-oriented models (Gamberini et al., 2009; Makssoud et al., 2015; Zha & Yu, 2014). Our approach considers that layout is fixed and the model is required to re-optimize layout-aware manner: Distances between stations and task-stations' constrains are known.

Task-station allocations constrains further increase the importance of allowing movements between stations: If two tasks are fixed at different stations and workers are not allowed to move between stations, then these tasks will require two different workers. If, on the other hand, we do allow workers to move (considering displacement times), it might be possible for only one worker to perform both tasks.

Movement times can further contribute when heavy or big pieces have to be assembled. Yazgan et al. (2011) present a model to solve a bus assembly line which needs more than one worker to perform special tasks named *common jobs or tasks*. Suppose one allows a worker to move between stations in an assembly line with common tasks: he/she might be allowed to spend part of his/her time at one station, dealing with individual tasks, then move to a colleague's station to perform the common task, and, finally, return to his/her station.

Based on the necessity of addressing particularities of real-world problems, this paper proposes a mathematical model to solve real-world ALBP problems along with the assignment of tasks and workers to stations for a given line layout. The model treats an assignment and a traveling salesman problem (TSP) for each worker of the line. The reason for addressing this particular feature is that further flexibility is given to the assignment of tasks. Furthermore, movements between workstations due to a different number of workers and stations or the presence of common tasks can be treated in the proposed unified model. Although both the addressed ALBP and TSP are NP-Hard problems (Baker, 1974; Papadimitriou, 1977), the model has been able to solve the generalized problem for real-world tested scenarios, as discussed within the results section. The model scalability is also indicated within the results section by an extensive set of tests based on standard datasets.

The paper is organized as follows: Section 2 presents the definition of the Traveling Worker Assembly Line Balancing Problem (TWALBP), modeled features, and assumptions. Section 3 presents a mixed-integer linear programming model for the extended version of TWALBP combining both balancing and TSP formulations. Section 4 contains the preprocessing procedures to eliminate variables and reduce the model's search space. Section 5 presents case studies and results for a simple version of TWALBP on adapted datasets, an illustrative example, and three real-world instances. A discussion section with the capabilities and limitations of the model is described in Section 6 while the conclusion is found in Section 7.

2. Problem Statement

The Traveling Worker Assembly Line Balancing Problem (TWALBP) is defined as follows: Given a set of tasks with deterministic processing times, precedence relations between tasks and deterministic movement times between stations one is asked to assign tasks to stations and stations to workers. The station-wise task assignment must respect the precedence relations. Each worker must, within the cycle time, perform the tasks assigned to the stations he/she is assigned to and move between said stations in a feasible cyclical pattern (no sub-cycles allowed).

Note that this is a *layout-aware* problem, not a layout-design problem, as the distances are all known *a priori* as parameters. The line shape is unimportant, provided it is known: straight, U-shaped, S-shaped, Multiple-U shaped, etc. The key information is that this layout is known, and the time-wise pair-wise distances between stations are known. TWALBP's main decision is no longer "Is the task t assigned to the station s?" as in regular balancing problems, but rather "Is the task t performed at the station s by the worker w?" and "Does the worker w move from station s_1 to station s_2?". In this sense this problem combines balancing decision variables to TSP's decision variables.

The case studies required further characteristics for the correct modeling of the real-world lines. Section 2.1 describes the extra features defined in an extended version of the problem (E-TWALBP). The Simple-TWALBP (S-TWALBP) is referred to the problem without the extended characteristics.

2.1. Shop-floor and Case Study Features

There are important practical features that are key to build the model described in the Section 3. The features of the described E-TWALBP are hereafter listed and briefly described:

- **Assignment Restrictions:** Due to equipment and ergonomic restrictions, some tasks can only be performed in a subset of stations for some problems. These restrictions can also be extended to Task-Worker capabilities, in the case of disabled workers or skill requiring tasks. Worker-Station assignment (zoning) constrains are useful to map the allocation of robots or stations with difficult access for disabled workers.

- **Common Tasks:** Some tasks require two or more workers to be performed. This can naturally imply on movements as some workers might be required to move to the station of a fellow worker in order to perform tasks, then return to his/her original station. These tasks, common to two or more workers, imply on key modeling differences discussed on the Section 3.

- **Automatic Tasks:** Some tasks are performed by machines, but require a worker to trigger them at a station (say by pressing a button). This means that the worker has to move in and out of that station, but the task's time was not added to the worker's time: It was performed by the machine while he/she performed other tasks and movements.

- **Different Processing Time per Station:** Tasks may be performed using multiple pieces of equipment or techniques. If the equipment of two workstations works at different rates, the station-wise processing time must be considered. Assembly lines with both manual and robotic labor can present significant differences on tasks' performance (an example is given in Section 5.2.3).

- **Fixed Time per Station:** Set-up times are considered given for each station regardless of the number of tasks performed, such as piece handling or positioning. These times are added to the worker's cycle time if he/she is assigned to the station. This is particularly significant if the workers are assigned to many stations, meaning they have to perform multiple set-up operations.

2.2. Model's Simplifications

Some particularities of the case study are modeled operating simplifications (hypothesis), due to overwhelming complexity that would otherwise arise. These simplifications are listed bellow:

- **Single Model:** In order to set aside unavoidable sequencing considerations, the mathematical model is restricted to single model ALB. Mixed-model cases are treatable if models present only small discrepancies in their processing time. Mixed-model instances could be modeled as a single model with task's time being the weighted average each model's task's time (Thomopoulos, 1970), but scheduling might become necessary.

- **Multiple Pieces per Station:** In one presented study case (Section 5.2.2), stations can process multiple pieces at a time. This is modeled as if task's duration on said station is divided by the number of pieces that could be processed simultaneously.

- **No Sequencing Required:** Common Tasks and Automatic Tasks could give rise to intricate sequencing considerations if multiple workers were set to perform multiple common and automatic tasks at multiple stations. A small number of common tasks, however, can be modeled without rigorously considering sequencing aspects (see Section 5.2.1).

The single model hypothesis is required in order to set aside sequencing considerations that would complicate the model significantly: Beside workpiece-model sequencing variables, each worker's cycle time constraint would require one to model multiple cycles. Furthermore, workers assigned to multiple stations might work with different models in the same cycle, as illustrated in Figure 2. Sparling & Miltenburg (1998) presented an improvement heuristic for the balancing of mixed-model U-lines. The pairs of models assigned to a crossover workstation are given from initial response. In a more recent work, Hamzadayi & Yildiz (2013) present a simulated annealing algorithm that has to determinate stations' assigned models for each of the explored solution. For TWALBP, once the assignment of workers to stations is part of the problem, a mixed-model TWALBP formulation would also require variables to map which models are occupying each station for any given time. Lopes et al. (2016) present a cyclical framework that can aid modeling such mixed-model variants.

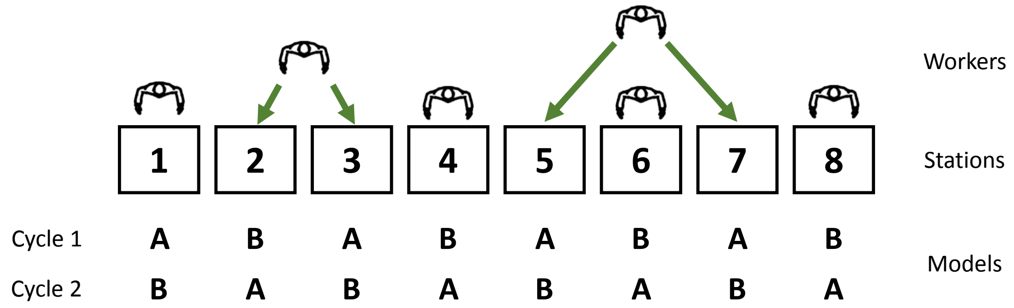


Figure 2: Possible line configuration for a mixed-model assembly line. Workers assigned to multiple stations may perform tasks on pieces of different models within a cycle. For instance, in Cycle 1, the worker assigned to stations 2 and 3 is responsible for models B and A while the worker assigned to stations 5 and 7 only performs tasks in pieces of model A. In the second cycle, the first multiple assigned worker has models A and B, while the latter works only with model B pieces.

The multiple pieces per station hypothesis divides the processing time by a station-wise fixed factor. This is required in order to take in account the station's capacity to process multiple pieces and not portrait the time the worker spend in the station as too high: Any delay caused by this consideration in one cycle would be compensated at the next ones.

The last consideration is valid because few of the tasks in the case studies were either automatic or common: five out of 81 in one case, three out of 121 in the later. Thus, given the model's simplifications and the specialized features to be addressed, Section 3 details the proposed mathematical model.

3. Mathematical model

The proposed model is designed to treat a 3 dimensional (task-worker-station) balancing problem together with a traveling salesman problem for each worker. The model presented in this section considers all the characteristics described in Section 2. The equations for the Simple-Traveling Worker Assembly Line Balancing Problem (S-TWALBP) are indicated in the text.

The following notations are used to describe the model:

	Indexes used in the model:	
t	index for the tasks:	1...NT
s	index for the workstations:	1...NS
w	index for the workers:	1...NW
n	necessary number of workers for a common task	

The following sets are used as data inputs for the model. The indexes show the dimensions of each parameter.

Further sets are used to define variables and restrictions. The sets only contain valid assignments and are created in a preprocessing phase based on the input data used. These sets are formally defined in Section 4.

The MILP formulation has variables from the assembly line balancing problem as well as from the traveling salesman problem. The balancing model is based on the formulation of Patterson &

	Problem's parameters:
DT_{ts}	duration time of each task for each workstation
$Prec_{t_1t_2}$	precedence relations for the task pair t_1-t_2: t_1 must precede t_2
$TSfeas_{ts}$	pairs of possible task-station allocations due to equipment restrictions
AT_t	automatic tasks: tasks that are performed by a machine rather than a worker
CnT_{tn}	common tasks t to be performed by n workers
FW_s	fixed workload of a station s due to setups or shared equipment with other lines
PpS_s	number of pieces per station produced in one cycle time of the station s
$MT_{s_1s_2}$	movement time spent for the dislocation between stations s_1 and s_2
$WSfix_{ws}$	list of fixed worker-station allocations
$WSfeas_{ws}$	list of possible worker-station allocations: zoning restrictions
$TWInc_{tw}$	list of tasks a worker cannot perform: used for ability restrictions

	Sets used in the model:
$Tasks$	set of all tasks
$Stations$	set of all stations
$Workers$	set of all workers
TWS	set of all task-worker-station feasible assignments
TS	set of all task-station feasible assignments
TW	set of all task-worker feasible assignments
WS	set of all worker-station feasible assignments
WSS	set of all movements of a worker w from station s_1 to station s_2
TWS_{cn}	set of all task-worker-station feasible assignments for the common tasks
TS_{cn}	set of all task-station feasible assignments for the common tasks
TW_{cn}	set of all task-worker feasible assignments for the common tasks
WS_{cn}	set of all worker-station feasible assignments for the common tasks
WSS_{cn}	set of all movements of a worker w as a side worker to help in a common task
T_{at}	set of all automatic tasks
T_{cn}	set of all common tasks
$\overline{T_{cn}}$	complement of T_{cn}
W_{free}	set of all workers without zoning or ability restrictions
TWS_u	represents the union of the sets TWS and TWS_{cn}
TS_u	represents the union of the sets TS and TS_{cn}
TW_u	represents the union of the sets TW and TW_{cn}
WS_u	represents the union of the sets WS and WS_{cn}
WSS_u	represents the union of the sets WSS and WSS_{cn}

Albracht (1975), while the movements for the workers are based on a traveling salesman problem (TSP) model by Miller et al. (1960). The model variables are represented with a v before the variable name.

The objective function is to minimize the cycle time of the assembly line, as stated by the Equation 1. A possible secondary objective, the total movement time minimization is also desirable. Dislocations are non-productive activities, and it is possible for two solutions with the same cycle time to have different total dislocation times. A solution with less movement time is preferred both because of ergonomic and safety factors. In order to also optimize in regard to this secondary objective, the model can be adapted to have an alternative goal function with two terms (namely Minimize $C_1 \cdot vCT + C_2 \cdot vMovTime$, with $C_1 \gg C_2$) or a two phase process can be used: the model is solved for the minimal vCT in the first run while the objective function for the second run is $vMovTime$ with vCT restricted to the optimal value found in the first run.

$$\text{Minimize: } Z = vCT \tag{1}$$

The constraints are organized in function groups. First we present the task-station assignments restrictions that are adapted from simple assembly line balancing models. The second set of restrictions define the assignment relations considering the workers allocations as well. The movements

	Model's variables:
vCT	cycle time of the line
$vWTime_w$	cycle time for each worker
$vSTime_s$	cycle time for each workstation
$vMovTime_w$	time used for each worker to move between stations
vTS_{ts}	is equal to 1 when task t is performed in station s, 0 otherwise one binary variable for each element of the set $TS \cup TS_{cn}$
vWS_{ws}	is equal to 1 when worker w is assigned to station s, 0 otherwise one binary variable for each element of the set WS
$vWScn_{ws}$	is equal to 1 when worker w performs a common task at station s, 0 otherwise one binary variable for each element of the set WS_{cn}
vTW_{tw}	is equal to 1 when task t is performed by worker w, 0 otherwise one binary variable for each element of the set $TW \cup TW_{cn}$
$vTWS_{tws}$	is equal to 1 when task t is performed by worker w at station s, 0 otherwise one binary variable for each element of the set $TWS \cup TWS_{cn}$
$vWSS_{ws_1s_2}$	is equal to 1 when worker w moves from station s_1 to station s_2, 0 otherwise one binary variable for each element of the set $WSS \cup WSS_{cn}$
$vWS0_{ws}$	is equal to 1 when the station s is the first station assigned to worker w one binary variable for each element of the set $WS \cup WS_{cn}$
$vOWS_{ws}$	position of each station s in the cycle each worker w performs one integer variable for each element of the set $WS \cup WS_{cn}$

from worker are modeled in restrictions shown in the last subsection.

3.1. *Task-station model restrictions*

The first set of restrictions is adapted from the assembly line balancing problem formulation from Patterson & Albracht (1975). Equation 2 is the occurrence restriction: each task has to be performed in a workstation. The restriction for the precedence relations between tasks is defined in ineq. 3 while ineq. 4 forces vCT to be limited by the most loaded workstation. The cycle time for each workstation is calculated by eq. 5. If the workstation produces more than one piece per cycle time, its cycle time is adjusted to the time necessary to produce one piece. Fixed workloads also have to account to the cycle time of a station. For the simple version of the problem, these extra terms may be left out.

$$\sum_{(t,s)\in TS} vTS_{ts} = 1 \qquad \forall\ t \in Tasks \tag{2}$$

$$\sum_{(t_1,s)\in TS} s \cdot vTS_{t_1 s} \leq \sum_{(t_2,s)\in TS} s \cdot vTS_{t_2 s} \qquad \forall\ (t_1, t_2) \in Prec \tag{3}$$

$$vStime_s \leq vCT \qquad \forall\ s \in Stations \tag{4}$$

$$vSTime_s = FW_s + \sum_{(t,s)\in TS} \frac{DT_{ts} \cdot vTS_{ts}}{PpS_s} \quad \forall\ s \in Stations \tag{5}$$

3.2. *Task-worker-station model restrictions*

Once the model also has to assign workers to tasks and stations, more restrictions are needed to model these relations. Equations 6 and 7 are the occurrence restrictions related to the task-worker assignment. Note that for the extended problem, a common task requires more than one worker to be assigned to the task.

$$\sum_{(t,w)\in TW} vTW_{tw} = 1 \qquad \forall\, t \in \overline{T_{cn}} \tag{6}$$

$$\sum_{(t,w)\in TW_u} vTW_{tw} = n \qquad \forall\, t \in T_{cn} \tag{7}$$

The sets of occurrence constraints defined in eqs. 2, 6, and 7 are not enough to model task-worker-station assignments. Further restrictions are necessary to link the binary variables with each other. Inequalities 8 - 10 represent a logic *and* constraint implying that a task is performed by a worker in a given workstation (TWS) only if the task is performed by such worker (TW) and the task is assigned to this workstation (TS). Inequalities 11 and 12 are complementary restrictions. They only allow the variables vTW and vTS to assume 1 if any of the $vTWS$ variables are also 1.

$$vTWS_{tws} \leq vTW_{tw} \qquad \forall (t,w,s) \in TWS_u \tag{8}$$

$$vTWS_{tws} \leq vTS_{ts} \qquad \forall (t,w,s) \in TWS_u \tag{9}$$

$$vTWS_{tws} \geq vTW_{tw} + vTS_{ts} - 1 \qquad \forall (t,w,s) \in TWS_u \tag{10}$$

$$\sum_{(t,w,s)\in TWS_u} vTWS_{tws} \geq vTW_{tw} \qquad \forall\ (t,w) \in TW_u \tag{11}$$

$$\sum_{(t,w,s)\in TWS_u} vTWS_{tws} \geq vTS_{ts} \qquad \forall\ (t,s) \in TS_u \tag{12}$$

When there are common tasks in a workstation, more than one worker-station assignment is necessary. We define workers as main workers and side workers or helpers. At a station with common tasks, the main worker can perform any task while side workers can only help with the common tasks. Because of that, the variable that determines the worker-station allocations for common and non-common tasks cannot be gathered in a single set. These differences imply that vWS and vWS_{cn} are used for the main and side worker allocations, respectively. The restrictions that bound vWS with $vTWS$ must account the different variable sets.

Inequality set 13 is the vWS adaptation of ineqs. 8 and 9. This inequality set assures that if a worker performs a task in a given workstation, the worker has to be assigned in such workstation. Depending on the set, the worker can be a main or side worker. For the non-common tasks, only the main worker allocations (vWS) is considered (ineq. 13a). For allocations which a worker could either be the main or the side worker, $vTWS$ must be linked to both the variables, as shown in ineq. 13b. Due to zoning restrictions or domain cuts that are explained in Section 4, it is possible that the allocation of a worker as a main worker is restricted while he/she could still help as a side worker in a given workstation. In such cases, ineq. 13c links $vTWS$ with vWS_{cn} variables.

$$vTWS_{tws} \leq vWS_{ws} \qquad \forall (t,w,s) \in TWS \setminus TWS_{cn} \tag{13a}$$

$$vTWS_{tws} \leq vWS_{ws} + vWScn_{ws} \qquad \forall (t,w,s) \in TWS\ \cap\ TWS_{cn} \tag{13b}$$

$$vTWS_{tws} \leq vWScn_{ws} \qquad \forall (t,w,s) \in TWS_{cn} \setminus TWS \tag{13c}$$

The main worker is defined as the worker assigned to the workstation for the normal tasks. Inequality 14 is necessary to restrict that only one worker can be the main worker in a workstation. Although the vTS and vTW occurrence restrictions force each task to be assigned to a station and workers respectively, that is not the case for the worker-station assignments. A worker would only be assigned to a workstation if it is necessary. In this way, some stations might stay unused in the balancing. The MILP model decides the stations' use, seeking for the best balancing conditions. This fact will be further exploited in Section 5.

Finally, to conclude with the task-worker-station assignment constraints, we need to impose that a main worker cannot be his/her own side worker, as shown in ineq. 15.

$$\sum_{(w,s)\in WS} vWS_{ws} \leq 1 \qquad \forall\ s \in Stations \tag{14}$$

$$vWS_{ws} + vWScn_{ws} \leq 1 \qquad \forall\ (w,s) \in WS \cap WS_{cn} \tag{15}$$

While a model for task-station assignments would need only eq. 2 to assure that every task is correctly assigned, restrictions 6 - 15 are needed to treat a task-worker-station assignment model with common tasks.

When only task-station assignments are considered, the cycle time would be limited by the most loaded workstation. On the other hand, when worker's activity time is accounted, the line's bottleneck can also be the most loaded worker (ineq. 16). As shown in Equation 17, the cycle time of a worker is the time he/she takes to perform the assigned tasks, the movement time and any fixed workload that the worker need to perform in the station, such as a set-up procedure, for instance. Each automatic task is assigned to a worker, but its processing time is not accounted in the worker's cycle time. For the simple version of the problem, only the processing time of a single model and the movement time are considered.

$$vWtime_w \leq vCT \quad \forall\ w \in Workers \tag{16}$$

$$\begin{aligned} vWtime_w = &\sum_{\substack{(t,w,s)\in TWS_u \\ t\ \notin\ T_{at}}} \frac{DT_{ts} \cdot vTWS_{tws}}{PpS_s} + \\ &+ vMovTime_w + \sum_{s\in Stations} (FW_s \cdot vWS_{ws}) \qquad \forall w \in Workers \end{aligned} \tag{17}$$

3.3. Worker's movement model restrictions

The used model for movements is based on a Traveling Salesman Problem model from Miller et al. (1960). The variable $vWSS_{ws_1s_2}$ is used to control the movement between workstations. Equation 18 assures that the movement time of each worker is equal to the sum of all dislocations. Variable $vWS0_{ws}$ determines in which workstation each worker is initially assigned. Every worker has to start in a workstation, as it is assured in eq. 19. Inequality 20 links the variables $vWS0_{ws}$ and vWS_{ws}: a worker can only start the movement from a workstation in which he/she has been assigned.

$$vMovTime_w = \sum_{(w,s_1,s_2) \in WSS_u} vWSS_{ws_1s_2} \cdot MT_{s_1s_2} \quad \forall w \in Workers \tag{18}$$

$$\sum_{(w,s) \in WS} vWS0_{ws} = 1 \qquad \forall w \in Workers \tag{19}$$

$$vWS0_{ws} \leq vWS_{ws} \qquad \forall (w,s) \in WS \tag{20}$$

For the movements, we have to account both worker-station assignments for main and side workers. Each worker has to perform a cycle between the stations that were assigned and return to the starting workstation (the workstation we assume is the origin of the movement). To complete the cycle, a worker would move once per assigned station. Equation 21 assures the number of movements of each worker is enough to cover all the cycle by summing WSS at the left side. The right side of the equation sums every allocation of a worker w as a main or side worker in any of the stations (generic s_3 and s_4 are used as station indexes).

$$\begin{aligned} \sum_{(w,s_1,s_2) \in WSS_u} vWSS_{ws_1s_2} &= \sum_{(w,s_3) \in WS} vWS_{ws_3} + \\ + \sum_{(w,s_4) \in WS_{cn}} vWScn_{ws_4} & \quad \forall\ w \in Workers \end{aligned} \tag{21}$$

Once workers can be either a main or a side worker in a station, the variables vWS_{ws} and $vWScn_{ws}$ have to be used in the same expression. A given pair (w,s) can be part of WS, WS_{cn} or both. We define vWS^+_{ws} as the sum of the variables for the main and side workers' assignments, if they exist, as follows: $vWS^+_{ws} = \sum_{(w,s) \in WS} vWS_{ws} + \sum_{(w,s) \in WS_{cn}} vWScn_{ws}$.

Equations 22 and 23 associate the movement variable $vWSS$ with the worker-station variable vWS^+. Equation 22 assures that if a worker is assigned to a workstation, a movement variable from this station to somewhere else has to assume 1. Analogously, eq. 23 forces a movement variable from somewhere to an assignment station to assume 1. Note that if a worker is only assigned to a single station, the variable for a reflexive movement ($s_1 = s_2$) would assume 1.

$$\sum_{(w,s_1,s_2) \in WSS_u} vWSS_{ws_1s_2} = vWS^+_{ws_1} \quad \forall\ (w,s_1) \in WS_u \tag{22}$$

$$\sum_{(w,s_1,s_2) \in WSS_u} vWSS_{ws_1s_2} = vWS^+_{ws_2} \quad \forall\ (w,s_2) \in WS_u \tag{23}$$

The order o of the station s to the worker w is the mechanism that eliminates sub-cycles for the TSP built within this balancing problem. The order of a station is 1 plus the order the previous station, except for the first station. Because we do not know through which stations the worker will have to go, we make that statement for every pair of stations s_p and s_s. In inequality 24, the number of stations NS acts as a *Big-M* factor that weakens the restriction for the exceptions, namely when the worker does not travel from s_p to s_s and when s_s is the starting station. It is not necessary to establish a particular value for o in the starting station, these variables are a mere mean to prevent sub-cycles without using the standard exponentially large set of restrictions of TSP model from Dantzig et al. (1954).

$$vOWS_{ws_2} \geq vOWS_{ws_1} + 1 - NS \cdot \left(1 - vWSS_{ws_1 s_2} + \sum_{(w,s_2) \in WS} vWS0_{ws_2} \right) \tag{24}$$
$$\forall (w, s_1, s_2) \in WSS_u \mid (w, s_1) \text{ and } (w, s_2) \in WS_u$$

3.4. Domain cut restrictions

The constraint represented in ineq. 25 works as a cut in the problem's domain. Within a cycle, it does not matter which node is assumed to be the initial one. Several equally good answers can be simplified if we assume each worker starts in the earliest station of the cycle. Inequality 25 forces that the value of $vWS0$ to be 1 for the earliest station in which a worker is assigned.

$$vWS0_{ws_1} \geq vWS0_{ws_2} - (1 - vWS_{ws_1}) \tag{25}$$
$$\forall w \in Workers, (w, s_1) \text{ and } (w, s_2) \in WS \mid s_2 > s_1$$

The precedence relations, as defined in ineq. 3, are only meaningful for workstations assignments. Once workers can move between stations, the model could allow a worker to be assigned to the first and the last station of an U-shaped line, for instance. A worker-based precedence applied to vWS would restrict the worker-station assignments in a way that the workers have to operate in line. The variable $vWS0$, on the other hand, only controls the first workstation of each worker. A precedence like restriction, ineq. 26, can be used for the ordering of workers. This way symmetrical solutions obtained by swapping workers are eliminated. Note that the restrictions are only applied for workers of the set W_{free}. Applying this cut for workers under zoning or ability restrictions could produce unfeasible solutions, so that the ordering is only applied to free workers.

$$\sum_{(w_1,s) \in WS} vWS0_{w_1 s} \cdot s \leq \sum_{(w_2,s) \in WS} vWS0_{w_2 s} \cdot s \quad \forall\, w_1, w_2 \in W_{free} \mid w_1 < w_2 \tag{26}$$

Inequalities 25 and 26 represent an intra and an inter-worker cycle ordering cuts, respectively. Although they are not functional restrictions, their presence are important to reduce processing time.

4. Preprocessing

In this section, we show how the sparse sets for the assignments are created. According to Sikora et al. (2015), the use of sparse sets in assembly line balancing problems with assignment restrictions can be a fundamental issue to reduce search space to just viable choices, contributing to the computational load reduction.

The first defined set is TW_{ini}, for the task-worker assignments. The index ini stands for an initial set, a further operation results in TW. As shown in eq. 27, TW_{ini} is a set for all task-worker combinations, unless it is stated that a worker cannot perform a task. The task-worker incompatibilities are parameters given by $TWInc$. These incompatibilities are useful to allow treating workers with different abilities or training, disabled, and robotic workers. When a line has both manual and robotic workers, the set $TWInc$ must map which tasks each kind of worker is unable to perform. Common tasks are considered to be performed by human workers, so that $TWInc$ must contain all common tasks and robots combinations.

$$TW_{ini} = \{(t, w) | t \in Tasks, w \in Workers, (t, w) \notin TWInc\} \tag{27}$$

The set WS_{ini}, for possible worker-station allocations, is $WS_{feas} \cup WS_{fix}$ with only two particular nuances: Firstly, we suppose robots cannot move between workstations: robotic workers have to be fixed to a workstation, only its feasible assignment position should be listed in WS_{fix}. The second nuance is that when we have a fixed worker, no other worker can have the option to be assigned to the fixed workstation. This assures that unnecessary variables are not created. Thus, the WS_{ini} set is either equal to the $WS_{feas} \cup WS_{fix}$ set, if such set has already been built according to the described nuances, or a filtered version of the same set, where the inadequate allocations have been eliminated. Further zoning restrictions are mapped in WS_{feas} and given to the model as a problem's parameter.

An additional domain cut can be applied. Setting ability restraints aside, all workers are supposed to be identical. As a result, multiple equally good answers can be obtained by swapping worker assignments. In order to reduce the answer multiplicity, a soft ordering is applied in the WS_{ini} set. WS^*_{ini} is defined in eq. 28 excluding the assignments of high numbered workers to the low numbered workstations.

$$WS^*_{ini} = \{(w, s) \in WS_{ini} | s \geq w\} \tag{28}$$

A task can be common (T_{cn}), automatic (T_{at}) or ordinary (T_{ord}) if it does not have another classification. The non-ordinary task assignments are given as a parameter in sets CnT_{tn} and AT_t, once these tasks have different restrictions applied to them. $TSfeas_{ts}$ is the union of the assignment possibilities of all types of tasks. The model must be informed which workstation has the necessary equipment for performing each task. Common and automatic tasks can have multiple capable workstations to be assigned, but usually such pieces of equipment are specialized and would only result in a single possible allocation. In the case of robotic workstations, task-station assignments have to match with the possibility of assigning a robot to perform the task.

The variable elimination process of Patterson & Albracht (1975) can be included in the formation of TS_{ini}. Equations 29 and 30 calculate the interval a task can be assigned according to its position on the precedence diagram. E_i is the earliest station in which task i can be assigned while L_i is the latest station. The duration time of a task i is represented as dt_i, while P_i and F_i stand for precedent and follower tasks, respectively. If a task's processing time varies according to the workstation, the minimal processing time must be used. Note that in the case of a surplus of workstations, the number of workstations (NS) weakens the variable elimination because the cycle time (CT) would probably be limited by a bottleneck worker. Note that if CT is unknown, an upper bound should be used.

$$E_i = \left\lceil \frac{dt_i + \sum_{j \in P_i} dt_j}{CT} \right\rceil \qquad \forall i \in \text{Tasks} \tag{29}$$

$$L_i = NS - 1 + \left\lceil \frac{dt_i + \sum_{j \in F_i} dt_j}{CT} \right\rceil \qquad \forall i \in \text{Tasks} \tag{30}$$

Equations 29 and 30, however, do not consider equipment restrictions, only the precedence relations are used. This intervals can be enhanced using the information of $TSfeas_{ts}$. The feasible earliest (latest) station of each task is defined by fE_i (fL_i) in the eqs. 31 and 32. These bounds are determined as the earliest and latest station a task can be assigned considering both the precedence diagram and the assignment restrictions.

$$fE_i = min(s \mid (i, s) \in TS_{feas}, s \geq E_i) \qquad \forall i \in Tasks \tag{31}$$

$$fL_i = max(s \mid (i, s) \in TS_{feas}, s \leq L_i) \qquad \forall i \in Tasks \tag{32}$$

Further domain reduction can be done by recalculating the bounds after assignment restrictions are considered. For instance, if the interval calculated for a given Task A is $[E_A, L_A]$, but Task A precedes a fixed Task B in workstation $S_B < L_A$, then the assignment interval for A should be updated for $[E_A, S_B]$. Equations 33 and 34 are expressions used to refine the bound taking into account restricted assignments of predecessors or successors. In Equation 33, the refined earliest station (rE_i) is either fE_i or the first station in which all its predecessors can already be assigned. The set TS_{ini} is then defined as the intersection of TS_{feas} and the interval $[rE_t, rL_t]$, showed in Equation 35.

$$rE_i = max(fE_i, max(fE_j)) \qquad \forall i, j \in \text{Tasks} \mid j \text{ preceeds } i \tag{33}$$

$$rL_i = min(fL_i, min(fL_j)) \qquad \forall i, j \in \text{Tasks} \mid i \text{ preceeds } j \tag{34}$$

$$TS_{ini} = \{(t, s) | (t, s) \in TS_{feas}, s \in [rE_t, rL_t]\} \tag{35}$$

Once all sets of pairs of entities are mapped, the task-worker-station set can be defined. TWS is the intersection of all subsets that build the assignments. Equation 36 shows how TWS_{tws} is built, while Equations 37 - 39 represent how the sets TW_{tw}, TS_{ts} and WS_{ws} are defined. TW differs from TW_{ini} in the cases in which task-worker assignments are unfeasible due to the combinations of unfeasibilities in worker-station and task-station assignments. For instance, if a task is fixed to a station and a worker is restricted to other stations, the worker would not be able to perform this fixed task. The same goes for the sets TS and WS.

$$TWS = \{(t, w, s) | (t, w) \in TW_{ini}, (t, s) \in TS_{ini}, (w, s) \in WS^*_{ini}\} \tag{36}$$

$$TW = \{\ (t, w) \mid (t, w, s) \in TWS_{tws}\} \tag{37}$$

$$TS = \{\ (t, s) \mid (t, w, s) \in TWS_{tws}\} \tag{38}$$

$$WS = \{\ (w, s) \mid (t, w, s) \in TWS_{tws}\} \tag{39}$$

The common tasks require more than one worker in the station to be performed. Therefore, it is also necessary to list assignment possibilities for the side workers. These possibilities have to be restricted to the common tasks only, a side worker cannot be able to perform any other task in that workstation. The set TWS_{cn}, as defined in Equation 40, is the set of common tasks in the station they can be assigned with every worker that is able to perform such tasks. Robotic workers are not considered to perform common tasks, so that $TWInc$ must restrict all common task-robotic worker combinations. TW_{cn}, TS_{cn}, and WS_{cn} are the projections of TWS_{cn} the same way TW, TS, and WS are derived from TWS.

$$TWS_{cn} = \{(t, w, s) | t \in T_{cn}, (t, s) \in TS_{ini}, w \in Workers, (t, w) \notin TWInc\} \tag{40}$$

Set WSS is the collection of all movement possibilities. It is build by allowing each worker to freely move between any workstation in which he/she can be assigned into. Equation 41 shows the set WSS construction, while eq. 42 is the equivalent for the movements to perform common tasks (WSS_{cn}). Eq. 42 considers movements from stations without common tasks to one with a common tasks (WS to WS_{cn}), transfers between stations with common tasks (WS_{cn} to WS_{cn}) and back to stations without common task (WS_{cn} to WS). Reflexive movements are not considered for common tasks, once they only make sense for non-moving workers at their main workstation.

$$WSS = \{(w, s_1, s_2) \mid (w, s_1) \in WS, (w, s_2) \in WS\} \tag{41}$$

$$\begin{aligned} WSS_{cn} = &\{(w, s_1, s_2) | (w, s_1) \in WS, (w, s_2) \in WS_{cn} && | s_1 \neq s_2\} \\ &\cup \{(w, s_1, s_2) | (w, s_1) \in WS_{cn}, (w, s_2) \in WS_{cn} && | s_1 \neq s_2\} \\ &\cup \{(w, s_1, s_2) | (w, s_1) \in WS_{cn}, (w, s_2) \in WS && | s_1 \neq s_2\} \end{aligned} \tag{42}$$

5. Results

The model functionalities are tested using adapted standard datasets and real case studies. The solver IBM ILOG CPLEX 12.51 was used along with a Core i7-3610QM (2.3 GHz) computer with 16.0 GB of RAM for all instances. Subsection 5.1 shows the performance of the model for instances of the adapted benchmark sets from Scholl (1999) and Otto et al. (2013) for the simple TWALBP and an illustrative example.

For the case studies in Subsection 5.2, real-world lines seen in automotive manufacturers and piece suppliers in the region of Curitiba, Brazil, are adapted into the instances presented in the case studies. The sum of different practical characteristics observed within these case studies motivated the creation of a more flexible model to efficiently solve such balancing problems. The E-TWALBP mathematical model was built in order to address specific restrictions of all the case studies that differed from the simple assembly line balancing problem. The Subsection 5.2 describe the problem's characteristics and how they are modeled.

5.1. *Simple Traveling Worker Assembly Line Balancing Problem Adapted Dataset*

In this section we show the positive effect of extra stations in the flexibility of the balancing maintaining the number of workers. To determine how performant the model is for heavily combinatorial cases, we propose Simple Traveling Worker Assembly Line Balancing Problem (S-TWALBP) datasets as adaptations of the SALBP-2 instances. In Subsection 5.1.1 the Scholl (1999) dataset is used and Subsection 5.1.2 is based on Otto et al. (2013) dataset.

5.1.1. *Model's performance on Scholl's dataset*

For this study, one extra workstation is added to every one of the 302 instances of the dataset, maintaining the same number of workers. This extra station allows one worker to be responsible for two workstations, moving between them as a result. Precedence relations must be respected considering the workstation order. For workers, however, precedence relations are not defined. The double assignment allows a worker to perform tasks in different regions of the precedence diagram: tasks that would, otherwise, not be assigned to the same worker due to precedence relations can now be assigned, but in separate workstations. This concept is different from the U-line Line Balancing Problem (U-line LB) defined from Miltenburg & Wijngaard (1994). The workstations assigned to a worker cannot always be considered one crossover workstation as defined by Miltenburg (1998) and, therefore, cannot be considered as a single combined station. The flexibility, however, is penalized with the time spent in the movements.

Not all assembly lines benefit from the flexibility of an extra station. The assignment of a worker in two workstations can only reduce the cycle time if the 1 on 1 assignments (each worker is assigned only to one station) contain significant idle time to pay off the movement costs. In order to identify instances with potential to be improved by the given model, we compared the optimum configuration for SALBP-2 obtained at `http://www.assembly-line-balancing.de` with a lower bound for the answer. The used bound is the $LC1$, described by Scholl & Becker (2006). The bound is given by: $LC1 = max\{DT_{max}, \lceil DT_{sum}/NW \rceil\}$, where DT_{max} is the duration time of the largest task and DT_{sum} is the sum of the duration time of all tasks. In addition, Equation 43 was used as a cut: once only one extra station is considered, so each worker w must be assigned to his/her natural position (station w) or the next one $(w + 1)$.

$$vWS_{w,w} + vWS_{w,w+1} \geq 1 \quad | \quad w \in Workers \tag{43}$$

Out of the 302 SALBP-2 instances in the dataset, 133 cases have a difference between the optimal answer and $LC1$ of one or more units of time, and are, therefore, considered in the study. Once the improvement depends on the movement time and the line physical configuration, one can obtain a lower bound for the cycle time considering the movement time to be null. Table 1 summarizes the results obtained for the 133 instances with a limit of one hour of processing, considering zero for the movement time between stations. The given problems are SALBPs integrated with a worker allocation problems, so they are expected to be hard to solve: only 49 of the tested instances were solved to optimality within 3.600 seconds of processing. However, for 61

Table 1: Results of the 133 adapted SALBP cases. The column $Case$ contains which precedence diagram was used for each instance. The column $No.Workers$ contains the number of workers of the instances with improvement potential. $\#OPT$ stands for the number of optimal answers found in a limit of 3,600 seconds. The column $\#B$ has the number of answers obtained whose cycle time is smaller than the optimal SALBP answer. $\#E$ stands for the number of cases in which the model obtained the same answer as the SALBP, while $\#W$ contains the number of answers worse than SALBP. The column $Imp.$ shows the average cycle time improvement for the cases better than the SALBP answer. Finally, the column $Avg.Var.$ contains the average number of variables for the set of problems.

Case	Tasks	Cases	No. Workers	#OPT	#B	#E	#W	Imp.	Avg. Var.
Arcus1	83	15	6-20	1	7	2	6	0.82%	7,587
Arcus2	111	12	14-19, 21-26	0	2	1	9	0.08%	24,755
Buxey	29	5	9-13	5	3	2	0	4.3%	2,544
Gunther	35	7	6-8, 10-13	7	7	0	0	3.54%	2,404
Hahn	53	7	3-9	7	7	0	0	4.37%	1,328
Lutz1	32	3	8-10	3	3	0	0	2.64%	1,588
Lutz2	89	2	18, 27	0	0	1	1	-	16,896
Lutz3	89	14	6-9, 11-13, 16-22	6	9	5	0	0.72%	7,273
Mukherje	94	21	5-25	12	18	1	2	1.38%	11,998
Sawyer	30	5	9-13	5	2	3	0	3.08%	2,838
Scholl	297	4	41, 43, 45, 49	0	0	0	4	-	168,256
Tonge	70	12	9-10, 12-13, 15-22	2	2	5	5	0.27%	9,228
Warnecke	58	15	11-12, 16, 18-29	1	1	4	10	0.70%	14,298
Wee-Mag	75	11	18-20, 23-30	0	0	8	3	-	32,280
Total		133		49	61	32	40	1.98%	16,613

instances, a cycle time better than the optimum for the SALBP was obtained. Notice that one hour of processing was not sufficient for the largest instances to achieve the SALBP optimal cycle time. The S-TWALBP optimal solution, however, will always be better or equal to the SALBP optimal answer.

5.1.2. Improvement Potential Based on Problem's Characteristics

Further tests were performed in the dataset provided by Otto et al. (2013) to measure the effects of the potential of an extra station on different problem structures. Otto et al. built instances based on real-world assembly line characteristics. The problems range from small (20 tasks), medium (50 tasks), large (100 tasks) and very large (1000 tasks). They observed two frequent precedence diagram characteristics on assembly lines: bottlenecks and chains. A bottleneck task is the only follower of multiple tasks and it also proceeds multiple tasks. On the other hand, chains of tasks are sets of activities that have a specific order. That is, these tasks have only one precedent and one follower. Finally, Otto et al. proposed three statistical distributions for the processing time of taks. These distributions are named *peak at the bottom*, *peak in the middle* and *bimodal*. The peak at the bottom distribution represents small tasks comparing to the cycle time, while the peak in the middle distribution produces tasks with processing time centered at one half of the cycle time. The bimodal instances mixes both: small tasks and big tasks. Otto et al. discussed that real-world assembly lines present either the peak at the bottom or the bimodal behavior. The peak in the middle distribution, even though it tends not to be found in practice, it produces the most challenging balancing instances.

The dataset is constructed with 525 instances for every problem size. The variations of the precedence diagrams and time distributions are used to create 21 types of problems, with 25 random variations for each type. The types of problems can be seen in Table 2: under the PD (precedence diagram) column, BN represents precedence diagrams with bottlenecks, CH represents chains and

$MIXED$ contains both, bottlenecks and chains. Different values for the graph ordering strength (OS) are used (0.2, 0.6 and 0.9). So that 7 different precedence diagram styles are provided. The processing time distribution (TD) is represented with the acronyms PB for peak at the bottom, PM for peak in the middle and BM for the bimodal distribution. Otto et al.'s instances were built as a SALBP-1 dataset to be solved for a standard cycle time of 1000 time units. For this study, we used the optimal number of workstations of the SALBP-1 instances to create the TWALBP-2 instances.

Table 2 contains the results for the 525 small instances with one extra station. In a similar reasoning used in Subsection 5.1.2, the movement time is considered to be null to evidence the improvement potential. Out of the 525 instances, 519 were solved to optimality within the limit of one hour and 382 presented a cycle time inferior to the SALBP optimal answer. The processing time distribution proved to be the most sensible effect on the improvement potential and the effective reduction on the cycle time. The instances with small tasks (peak at the bottom) are easier to balance with low idle time. On average, the distance between the SALBP optimal answer and the lower bound $LC1$ ($max\{DT_{max}, \lceil DT_{sum}/NW \rceil\}$) is only 0.58% (column %Pot.). Although the peak at the bottom instances contain the least improvement potential, by adding only one extra station, 95% of this potential can be achieved (column % Imp. ratio). An opposite behavior occurs with the peak in the middle instances. Tasks with processing time close to the middle of the cycle time are difficult to be matched together, resulting in very high idle times. The average improvement potential of such instances is 13.77%, but due to the difficulty of matching tasks, one extra station can only contribute to 5.74% of the available improvement potential. Finally, the best results occurred for the bimodal (BM) instances. With most varied task times, the addition of one extra station achieved the average improvement of 1.05% in the cycle time of these instances. Out of the average improvement potential of 1.85%, 56.37% of the distance to the lower bound can be reached allowing one worker to change stations.

Further conclusions can be drawn from the structured dataset of Table 2: the higher the ordering strength (OS), the higher is the potential and effective improvement on the cycle time by the proposed approach. Highly constrained precedence diagrams may produce unavoidable idle time, due to the lack of combining options for tasks. Moving workers can, therefore, be most valuable in such conditions. Furthermore, the more constrained instances were solved significantly faster.

In a similar reasoning used for Table 2, Table 3 presents the results for the medium instances (50 tasks). These problems have proven to be harder to solve, out of the 525 instances tested, 139 cases were solved to the optimality within the time limit of one hour. Although the cases were harder to solve comparing to the small cases, the same conclusions on the precedence diagrams and time distributions can be drawn. The potential and effective improvement depends mainly on the time distribution of tasks. The relative difficulty of different types of problems can be observed by the number of solved instances. The peak at the bottom instances, which are easier to solve for SALBP in comparison to the peak in the middle and bimodal, are also the easiest instances

Table 2: Results for the small instances adapted from Otto et al. (2013) dataset. The column PD stands for the form of the precedence diagram: BN represents diagrams containing Bottlenecks, CH contains Chains and MIXED both structures, while the column OS stands for the ordering strength of the graph. $\#PC$ contains the number of cases with improvement potential (difference between the SALBP answer and LB1). $\%Pot.$ represents the average improvement potential from the SALBP optimal answer. The column Time shows the average amount of seconds needed to solve the instances. $\#OPT$ stands for the number of optimal answers found in a limit of 3,600 seconds. The column $\#IC$ has the number of answers obtained whose cycle time is smaller than the optimal SALBP answer. $\%Imp.$ stands for the average improvement obtained from the SALBP answer. Finally, column $\%Imp.ratio$ contains the ratio between the improvement obtained with one extra worker and the total improvement potential from SALBP.

PD	OS	TD	#PC	% Pot.	Time (s)	#OPT	#IC	% Imp.	% Imp. ratio
BN	0.2	PB	15	0.08%	0.84	25	15	0.08%	100%
BN	0.6	PB	21	0.37%	1.18	25	21	0.36%	96.7%
CH	0.2	PB	11	0.05%	5.86	25	10	0.05%	91.7%
CH	0.6	PB	24	0.55%	0.94	25	22	0.51%	93.9%
MIXED	0.2	PB	3	0.01%	1.05	25	3	0.01%	100%
MIXED	0.6	PB	23	0.43%	1.05	25	23	0.41%	94.2%
MIXED	0.9	PB	24	2.55%	0.5	25	23	2.27%	88.9%
BN	0.2	PM	25	13.42%	1115.6	23	10	0.18%	1.3%
BN	0.6	PM	25	14.29%	361.6	25	19	1.06%	7.4%
CH	0.2	PM	25	11.87%	1053.6	25	10	0.35%	2.9%
CH	0.6	PM	25	14.24%	256.5	25	22	1.29%	9.1%
MIXED	0.2	PM	25	12.44%	1572.7	22	6	0.35%	2.8%
MIXED	0.6	PM	25	12.89%	309.1	25	15	1.05%	8.2%
MIXED	0.9	PM	25	17.22%	17.4	25	24	1.46%	8.5%
BN	0.2	BM	22	0.39%	117.6	25	20	0.21%	53.4%
BN	0.6	BM	25	1.55%	7.35	25	24	0.98%	62.6%
CH	0.2	BM	25	0.51%	235.9	25	22	0.3%	58.6%
CH	0.6	BM	25	2.14%	3.63	25	25	1.24%	58.7%
MIXED	0.2	BM	24	0.46%	591.2	24	20	0.2%	44.7%
MIXED	0.6	BM	25	1.83%	4.49	25	25	1.15%	63.3%
MIXED	0.9	BM	25	6.1%	0.74	25	23	3.24%	53.2%
Summary		PB	121	0.58%	1.6	175	117	0.53%	95.04%
		PM	175	13.77%	669.5	170	106	0.82%	5.74%
		BM	171	1.85%	137.3	174	159	1.05%	56.37%
Total/Avg.			467		269.5	519	382		

for the TWALBP. While the peak in the middle instances represent the most challenging class of problem for both: SALBP and TWALBP.

Tables 2 and 3 show that allowing workers to move between workstations can produce better balancings by adding flexibility to the allocation of tasks to workers. The more restricted the precedence diagram (or the task-station assignments) is, the greater is the effect of such flexibility. Not only precedence relations can be softened by using extra stations, but also practical restrictions such as fixed or assignment restricted tasks can also benefit from this reasoning.

5.1.3. An Illustrative Example

From the tested instances, we chose the precedence diagram of Hahn with 7 workers (from Scholl's dataset, as indicated in Section 5.1.1) for a further analysis. This case represented the best improvement potential in the cycle time (14%) when one extra station is allowed. Furthermore, the instance is small enough (53 tasks) to be solved to optimality in few seconds.

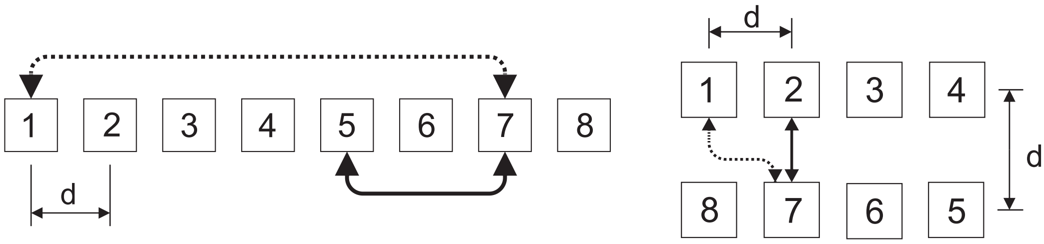
For the Hahn-7 instance, two configurations are defined to test the effect of the movements: a straight-line and a U-shaped line, represented in Figures 3a and 3b. The scenarios are solved for different degrees of movement times. We considered that the distance between adjacent stations increases linearly in the straight-line. For the U-shaped line, the Manhattan distance is used. The matrix showed in Table 4 contains the multiplication factors for the distance between stations. The temporal distance between two adjacent stations (d) is used as the control parameter.

Table 3: Results for the medium instances adapted from Otto et al. (2013) dataset. The column PD stands for the form of the precedence diagram: BN represents diagrams containing Bottlenecks, CH contains Chains and MIXED both structures, while the column OS stands for the ordering strength of the graph. $\#PC$ contains the number of cases with improvement potential (difference between the SALBP answer and LB1). $\%Pot.$ represents the average improvement potential from the SALBP optimal answer. The column Time shows the average amount of seconds needed to solve the instances. $\#OPT$ stands for the number of optimal answers found in a limit of 3,600 seconds. The column $\#IC$ has the number of answers obtained whose cycle time is smaller than the optimal SALBP answer. $\%Imp.$ stands for the average improvement obtained from the SALBP answer. Finally, column $\%Imp.ratio$ contains the ratio between the improvement obtained with one extra worker and the total improvement potential from SALBP.

PD	OS	TD	#PC	% Pot.	Time (s)	#OPT	#IC	% Imp.	% Imp. ratio
BN	0.2	PB	0	0%	1483.3	16	0	0%	0%
BN	0.6	PB	11	0.09%	1371.5	19	10	0.08%	86.36%
CH	0.2	PB	0	0%	2340.1	13	0	0%	0%
CH	0.6	PB	21	0.17%	2173.3	13	15	0.11%	65%
MIXED	0.2	PB	0	0%	1729.2	16	0	0%	0%
MIXED	0.6	PB	18	0.1%	2744.1	9	5	0.02%	26.09%
MIXED	0.9	PB	25	2.18%	22.4	25	24	1.42%	65.13%
BN	0.2	PM	25	7.69%	3600	0	0	0%	0%
BN	0.6	PM	25	11.62%	3600	0	0	0%	0%
CH	0.2	PM	25	7.7%	3600	0	0	0%	0%
CH	0.6	PM	25	11.28%	3600	0	0	0%	0%
MIXED	0.2	PM	25	8.68%	3600	0	1	0.03%	0.37%
MIXED	0.6	PM	25	11.76%	3600	0	0	0%	0%
MIXED	0.9	PM	25	16.92%	3541.5	1	8	0.31%	1.82%
BN	0.2	BM	10	0.05%	3600	0	0	0%	0%
BN	0.6	BM	25	0.44%	3517.7	1	4	0.02%	4.72%
CH	0.2	BM	20	0.09%	3600	0	0	0%	0%
CH	0.6	BM	25	0.81%	3600	0	6	0.07%	8.25%
MIXED	0.2	BM	9	0.04%	3590.4	1	0	0%	0%
MIXED	0.6	BM	25	0.55%	3600	0	5	0.07%	12.21%
MIXED	0.9	BM	25	4.6%	66.3	25	24	2.17%	47.23%
		PB	75	0.36%	1694.8	111	54	0.23%	34.65%
Summary		PM	175	10.81%	3591.6	1	9	0.05%	0.31%
		BM	139	0.94%	3082.1	27	39	0.33%	10.34%
Total/Avg.			389		2789.5	139	102		

Figure 4 shows the obtained cycle time for every tested value of the distance between adjacent stations. The distance is measured in a percentage of the cycle time of the SALBP instance (2336 Time Units) rounded to the nearest integer, that is, we consider a temporal distance. For instance, if we take 10% of the cycle time as the distance between stations, moving from adjacent stations would take 234 time units (rounded from 10% of a cycle time of 2336 time units). Note that the cycle time does not increase linearly with the augmentation of the stations' distances. For every given distance, the model considers the movement time in the balancing of the tasks, distributing the workload evenly. The U-shaped configuration obtained better results once stations are closer to each other in comparison to a straight line. The movements from the beginning to the end of the line produced improved answers for the U-shaped configuration. These movements, however, represent large displacements on a straight line. For this reason, the smaller distances between stations in U-lines enables a better workload balancing.

When movement times are low compared to the cycle time, long displacements can be beneficial to the output of a line. According to Table 5, when the distance of a station is up to 1% for the straight line and 6% for the U-line, the optimal assignment is to have a worker moving from station 1 to 7 (Configuration 1 in Figure 3a and Figure 3b). When the temporal distances between adjacent stations vary from 2 to 14% of the cycle time, it is worth having a worker assigned to both stations 5 and 7 in the straight line (Configuration 2 in Figure 3a). For greater values, the wasted time in



(a) Straight line layout. (b) U-shaped line layout.

Figure 3: Tested configurations for the Hahn problem with 7 workers and 8 workstations. The distance between adjacent stations (d) is the parameter observed in the tests showed in Figure 4. The arrows show the optimal allocation of workers that perform tasks in two stations. For small values of d, the doted line represent that one worker is assigned to stations 1 and 7 (Configuration 1). The continuous arrow shows the assignment for greater values of d (Configuration 2). The intervals for d are described in Table 5.

Table 4: Matrix of the distance between stations (MT_{ij}) for the straight and U-shaped lines for the Hahn-7 instance. The distance is the value of the matrix multiplied by the distance parameter d.

	Straight Line									U-shaped Line							
MT_{ij}	1	2	3	4	5	6	7	8	MT_{ij}	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7	1	0	1	2	3	4	3	2	1
2	1	0	1	2	3	4	5	6	2	1	0	1	2	3	2	1	2
3	2	1	0	1	2	3	4	5	3	2	1	0	1	2	1	2	3
4	3	2	1	0	1	2	3	4	4	3	2	1	0	1	2	3	4
5	4	3	2	1	0	1	2	3	5	4	3	2	1	0	1	2	3
6	5	4	3	2	1	0	1	2	6	3	2	1	2	1	0	1	2
7	6	5	4	3	2	1	0	1	7	2	1	2	3	2	1	0	1
8	7	6	5	4	3	2	1	0	8	1	2	3	4	3	2	1	0

the displacement is greater than the flexibility benefit. For the U-line, distance parameters from 7% up until 44% result in the optimal assignment of a worker in stations 2 and 7 (adjacent in a U-line, Configuration 2 in Figure 3b). Only for values greater than 45% the optimal solution is to have one worker in one station.

The Hahn-7 instance has a potential of improvement in the cycle time of 14% with one double worker assignment. For other cases, however, the improvements might be limited. Out of the 61 instances from the first dataset for which the cycle time reduced (Table 1), the average improvement in the cycle time was 1.98%.

For SALBP, the flexibility given by the movements soften the precedence relations. A worker can perform tasks in different regions of the precedence diagram, as exemplified in Figure 1. This flexibility is of even bigger importance when task assignments are further restricted to the allocation of expensive equipment, which are common in real-world assembly lines.

5.2. Real-World Assembly Lines

In this section we present real-world assembly line problems solved using the model for the Extended Traveling Worker Assembly Line Balancing Problem (E-TWALBP). The problems' data can be found in Appendix A. The three cases were solved to optimality in less than 3 minutes.

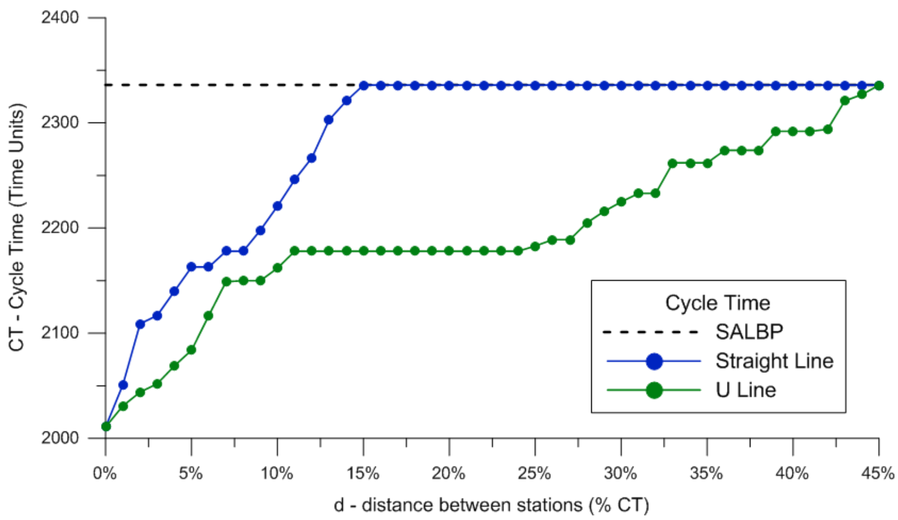


Figure 4: Cycle time for the optimal configuration for the Hahn instance with 7 workers and 8 stations. The distance between adjacent stations (d) varies from 0% of the cycle time to 45%.

Table 5: Resulting line configurations of the Hahn-7 instance for intervals of distance between stations (d). The movements and allocations are illustrated in Figure 3. For small values of movement time, longer movements are feasible (1-7). With longer movement times, the optimal answer shifts for closer assignments (5-7 and 2-7). The configuration 1 on 1 (each worker is assigned only to one station) is obtained for high values of movement times, that is, when the time used to dislocate is greater than its benefit.

Layout		Small Movements (allocation)	Big Movements (allocation)	1 on 1
Straight Line	d	0% to 1% (1-7)	2% to 14% (5-7)	$\geq$ 15%
U-Line	d	0% to 6% (1-7)	7% to 44% (2-7)	$\geq$ 45%

The low processing time is only reachable using the preprocessing 4 and the domain cut of the inequalities 25 and 26 (Subsection 3.4). The third case, in particular, took more than one hour to solve without the cuts and could not reach optimality with such computer configuration without the preprocessing. Thus, the preprocessing step, which generates sparse sets, associated with the domain cut constraints, were fundamental issues applied to reduce the computational burden.

5.2.1. Truck Cabin Assembly

The first example is an assembly line of truck cab with 15 workstations, 9 human workers and a robot. Two model variations account for 150 operations gathered in 81 indivisible tasks. The two models are very similar, over 80% of tasks have identical duration time. The simplest model, whose task durations are always lesser or equal to the more complex model, has a small production rate. Therefore, we treat the line as a single-model assembly line for the more complex product variant.

The line is responsible for the welding of the cabin parts. Many of these procedures need special tables to lock and hold the pieces in the correct position for the assemblage. Each working table is considered to be a workstation. The number of stations is then justified by the need of specialized equipment. Out of the 81 tasks, 44 need a specific equipment and are, therefore, fixed or restricted to the stations that contain such tools. Once the number of workstations is greater than the number of line operators, they have to dislocate between stations. Although the movement time is not productive, it has to be taken into account in the calculation of the cycle time.

Some of the operations are defined as common tasks. Such tasks need to be performed by more than one worker at the same time. Handling heavy or big pieces may require the action of extra workers. The positioning of truck bench structures is an example of a common task. In another case, a welding procedure is performed with the help of a platform. In all procedures using this equipment, a worker is necessary to control the platform movement during the welding.

Although most of the tasks are performed by manual labor, the line also has a robot. A mixed-labor line presents some complications: robots cannot move between workstations within the studied case and their capacity of performing tasks might also be different. In this line, welding procedures that unite pieces must be performed by human workers in specific tables while welding points used to reinforce the structure can be assigned to either a human or the robotic worker. The robotic worker can perform tasks with a lower processing time and is placed in the end of the line. As described in Section 4, a robot can be modeled as a worker fixed to a station while its capacity must be mapped in $TWInc$. The tasks that need more than one worker are modeled as common tasks fixed to the workstation in which the necessary equipment is available.

The objective of this case study is to minimize the cycle time. The optimal answer (214.65 TU) reduced the cab processing time in 11.3% (Appendix B, Table B.2) in relation to the original case alongside a new and simpler zoning setup that minimized movement times. The movement of each worker is a cycle controlled by a TSP formulation. As an example, in Table 6, we can see that Worker 1 moves from station 1 to 2, 2 to 4, and 4 to 1: the cycle of stations is 1-2-4. In particular, tasks 25, 50, 52, and 58 were common tasks (Table A.4) and Table B.1 also highlights this task-worker-station assignment. For instance, according to Table 6, Worker 4 is allocated as the main worker at station 7 and Worker 6 is the side worker for the common task. Further details about obtained results are presented in Appendix B.

Table 6: Movements performed by the workers in Study Case 1. A Worker W moves from station S_p to station S_s.

W	1	1	1	2	2	3	3	3	4	5	5	6	6	6	7	7	7	8	9
Sp	1	2	4	3	6	5	12	13	7	8	12	7	9	12	10	11	12	12	14
Ss	2	4	1	6	3	12	13	5	7	12	8	9	12	7	11	12	10	12	14

The obtained answer had Worker 6 as side helper in both stations 7 and 12. This double assignment might cause waiting times and, therefore, require scheduling if they cannot be synchronized. A simple alternative is to add Equations 44 - 46 as constraints in the problem. Equation 44 assures that workers only act as side helpers in one workstation. Equations 45 and 46 are also useful to this case study. They only allow a side-worker to perform one common task per station. Equation 45 is active if the worker cannot be the main worker, and Equation 46 only allows the main worker to perform all common tasks in the station.

$$\sum_{(w,s)\in WScn} vWScn_{w,s} \leq 1 \qquad \forall w \in Workers \quad (44)$$

$$\sum_{(t,w,s)\in TWScn} vTWS_{t,w,s} \leq vWScn_{w,s} \qquad \forall (w,s) \in WScn, (w,s) \notin WS \quad (45)$$

$$\sum_{(t,w,s)\in TWScn} vTWS_{t,w,s} \leq vWScn_{w,s} + NT \cdot vWS_{w,s} \qquad \forall (w,s) \in WScn, (w,s) \in WS \quad (46)$$

By applying the restrictions 44 - 46 a slightly inferior balancing with 215.3 TU is obtained (in comparison with 214.65 TU initially obtained). This response, however, is less prone to need scheduling.

5.2.2. Engine Block Machining

The second real line case is from an automotive parts machining company. The engine block section is composed by 4 machining centers that perform 41 tasks at a cycle time of 140.5 TU (Time Units). The centers are divided in 3 stations, because, in the last station, two centers work in parallel. Some machines have a double spindle and can, therefore, produce two pieces simultaneously.

The centers have different capabilities. Because of precision and access angles, there are operations that are performed exclusively in one station, while others can be allocated in more stations or even in all of them. The machining time depends on the cutting speed and depth of the centers. So, the duration of a task depends on which station it is performed.

The process duration does not depend on human influence. All machines have two pallets. After the procedure is finished, the center spins the pallets, and starts automatically with the new piece. A worker fills the pallets while the machines are working, so the cycle time is defined only by the machine operation time. Furthermore, due to position precision concerns, some tasks have to be performed in the same workstation. This restriction can be easily solved by merging these tasks in a single task with a duration time equal to the sum of the individual tasks' duration.

Within the considered modeling approach, the machines can be considered as workers fixed to a workstation. This way, the worker-station assignment is trivial and the balancing problem only has to solve the task-station assignments. The time it takes from a machining center to spin the pallet and to start its operation can be modeled as a fixed workload for the station. This way, different setup times from each machine can be easily contemplated in the balancing.

This is an ALBP-2 problem, the number of machines is given and the objective is to minimize the cycle time. Previously, the balancing for the machining centers was defined heuristically by assigning a task to the fastest workstation for that operation. The optimal balancing obtained by the proposed model improved the cycle time in 9.4% (Appendix B, Table B.4) resulting in an optimal answer that contradicted the heuristic solution. Some tasks that require almost twice as much time in a given station are nevertheless reassigned to balance the workload, as further indicated in Appendix B, Table B.3.

5.2.3. Gearbox Assembly

In the third case study, a single model of gearbox is assembled in a U-shaped line of 23 workstations and initially 20 workers including 2 robotic workers. A total of 1300 activities are divided into 121 groups of tasks. The group of tasks gathers operations that cannot be separated.

The need of specific tools for the assembly justifies the high number of workstations. Gears and bearings must be fixed under pressure in special equipment for each kind of operation. Again, movements from workers that are assigned to more than one station have to be taken in account in the cycle time.

In this case, both automatic and robotic specific tasks are present. Automatic tasks are operations in which a worker must inspect the piece and start the operation, while the rest of the task is performed by a machine. While the automatic task is performed, the worker is free to work in other tasks. Robotic specific tasks, on the other hand, must be assigned only to robotic stations due to precision restrictions. For this case, the robots cannot be easily adapted to perform other tasks, so their allocation is supposed to be fixed. Eliminating this tasks from the model, however, could create solutions that violate the precedence diagram. Although these tasks are fixed, they are important to define the process as it is in the real line.

The line operated clearly unbalanced with the cycle time of 1540.6 TU. The objective of this case is to find an assignment that results in a cycle time of 1350 TU, in accordance with the other lines in the industry. A further objective is to minimize the number of workers in the line: an ALBP-1 problem limited by 1350 TU. Although the model is built for ALBP-2 problems, iterative runs with different number of workers can be used to solve this problem.

It is possible to obtain an optimal cycle time of 1345 TU with at least 17 workers including the 2 robots, while the answer for a line with 16 workers exceeds the necessary cycle time. Every worker elimination implied in increments of movements from another worker to cover for the lacking worker. Due to the specialized equipment, the task allocations do not present much flexibility. On the other hand, by having more stations than workers the operational time of each worker can be more evenly balanced.

The Figure 5 shows how the line was implemented before the study. Workers were either responsible for a single station or pairs of adjacent stations. In Figure 6, the model's answer with 17 workers shows slightly longer movements. They are, however, necessary to reach the optimal cycle time due to precedence relations and the tasks' indivisibility constraint. The model considers the assignments of each station to decide which ones are the better options to include in a worker's route. The model explores the U-shaped line to assign workers to low-loaded workstations that are close to each other. Note that a worker performs tasks from the two internal sides of the U-line (Stations 5 and 9) while other is responsible for the beginning and the end of the line (Stations 1 and 23). The model also left one station unused (Station 13), which is indeed a viable practical option.

The Figure 7 shows the workloads associated to each station (Fig. 7a) and to each worker (Fig. 7b). Due to assignment and precedence constrains, the station-wise balancing is rather odd,

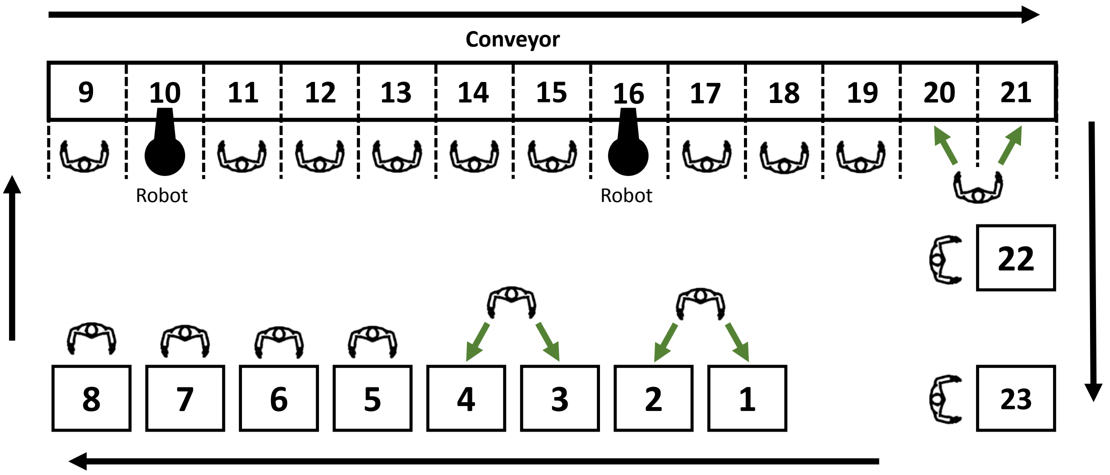


Figure 5: Line disposition and worker assignments in the line's original balancing (18 workers and 2 robots).

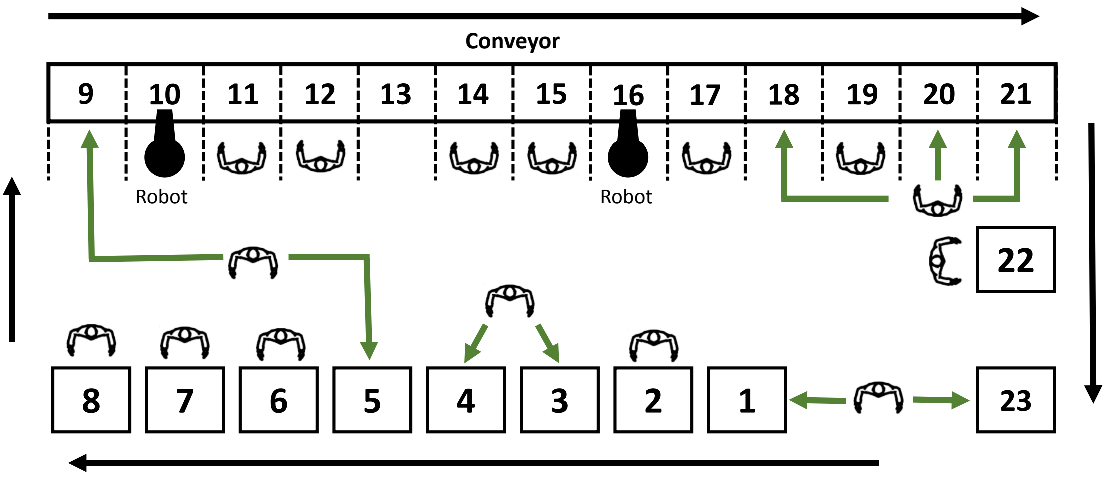


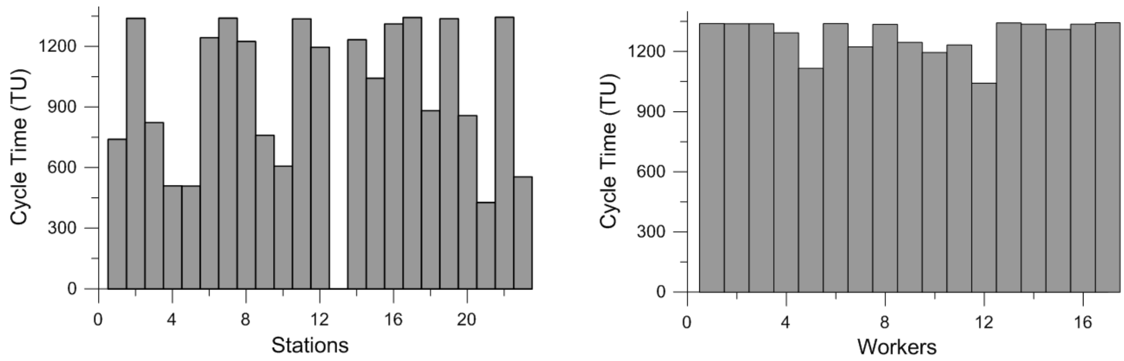
Figure 6: Line disposition and worker assignments given by the model using 15 workers and 2 robots.

but the worker-station allocation allows for an absorption of the observed differences. Regarding only the workstations, the balancing seems counter-intuitive, but the solution is perfectly logical looking at the workers' workload distribution. Further details about obtained results are presented in Appendix B, Tables B.5, B.6, and B.7.

5.3. Reduction of Variables by Preprocessing Techniques

A SALBP has a domain of boolean variables that determines which task is performed in each station. Every task can be allocated in any of the stations. The number of total variables is then equal to the number of station times the number of tasks ($NT \cdot NS$).

In the case where the number of workers and the number of stations differ, the workers must also be allocated in the stations. In this case, the number of possible allocations is the product of the number of tasks, workers, and workstations ($NT \cdot NW \cdot NS$).



(a) Cycle time for the stations

(b) Cycle time for the workers

Figure 7: Cycle time distributions obtained by the model.

The proposed mathematical model's preprocessing procedure eliminates the impossible task-worker-station allocations. This impossible allocations are results of equipment restrictions, worker zonings, incompatibilities of tasks, and others.

In the second study case (Subsection 5.2.2), the tasks are only performed by the machining centers, which are already allocated in a sequence. Therefore, to calculate the total task-worker-station boolean combinations it is just necessary to multiply the number of tasks with the number of centers. For the other cases, one must consider the three-term multiplication.

Once real applications have several restrictions related to cost of moving equipment and changing workstations, a balancing problem that takes those into account has less degrees of freedom to allocate tasks. Equipment, zoning, and ability restrictions can restrain the number of possible outcomes. The Table 7 shows the difference of the number of all possible combinations and the ones used after the preprocessing for the case studies proposed in Sections 5.2.1 to 5.2.3. Not only less variables are created, the number of restrictions is proportionally reduced. Thus, the detailed modeling of such real operational conditions contributed to prune the search space. The preprocessing shortens the time required to solve instances extending the application of models to more complex and bigger problems (Battaïa & Dolgui, 2012).

Table 7: The number of variables for all possible task-worker-station allocations and the equivalent after the preprocessing.

Case No.	All Combinations	Preprocessed Comb.	% Reduction
1	12150	828	93%
2	123	52	58%
3	55660	5084	91%

6. Discussion

The model (in particular its extended version, E-TWALBP) was created to treat particularities found in real lines that were not treatable using SALBP based models. Due to several restrictions of specialized or heavy machines, the assignment options for tasks are sometimes severely restricted. This lack of flexibility (strengthened by the precedence relations) might limit the balancing's quality since little interchange of tasks may be possible. If, however, there are more workstations than workers, additional degrees of freedom arise for the assignment of tasks and stations to workers. Tasks have to obey a precedence diagram that relates to their workstation allocation. Workers, on the other hand, can perform tasks in earlier stations and move to a latter point in the line within one cycle time. Although movements are unproductive time, this possibility might allow better balancing and pay off the time spent. Once the obtained results from the assembly line balancing problem and the traveling salesman problem can be both measured in time units, the model can be used to calculate the trade off and decide if and which movements bring advantages to the task assignment. Furthermore, the capacity to treat movements allows the model to describe common tasks, that are shared in a station between workers who can perform tasks at other stations. Taking these movements into account is a key point for the practical cases.

The model treats common and automatic tasks under the assumption that no sequencing is needed. A balancing model accounts which tasks are performed in which station, but not the order in which they are performed. The cycle time is calculated as the sum of the tasks and movements performed by the workers. If several common or automatic tasks are present, the waiting time may also be important in the determination of the cycle time. For these cases, a simultaneous balancing-sequencing model should also be considered. The model presented here supposes no waiting times are present. For instance, that a side worker would be able to help when a common task is to occur.

The most complex case study in terms of special tasks (Section 5.2.1) had four common tasks and one automatic task (out of 81 tasks): these were exceptions that had to be modeled in order to describe the problem. A close inspection of the model's answer reveals that sequencing would not be problematic for the provided answers: Cyclical schedules that respect the cycle time are possible for every worker. A balancing and scheduling formulation would require remodeling the problem in terms of additional scheduling variables for when each worker enters and leaves each station and when does he/she start and finish each task. This is pointed as a direction for further works, which can be based on simultaneous balancing-sequencing (or balancing-scheduling) approaches such as Öztürk et al. (2013, 2015). The proposed model did present feasible answers for balancing task allocation and worker displacement for the real case studies. The common and automatic tasks are taken in account and the model outputs allows one to provide worker-wise optimal cycles.

Tests with the simpler version of the model, applied to adapted datasets, have shown that in some cases it is possible to out-perform SALBP answers. The illustrative example (Subsection 5.1.3), in particular, allowed us to verify that this cycle time improvement is layout dependent. U-lines have significant advantages over straight-lines by bringing the "different regions" of the precedence diagram physically closer to one another. The displacement times are scarcely studied, and, therefore, this trade-off between flexibility and movement time is often overlooked. The presented model, on the other hand, allows an exact evaluation of the trade-off by combining a simple balancing model with traveling salesman sub-problems.

7. Conclusion

In this paper a line balancing problem is modeled in terms of task assignment to workers and stations, with a built-in TSP formulation to take worker movements between stations in account. The possibility to move between workstations gives the model more degrees of freedom to find solutions, as some precedence relations can be relaxed, as illustrated in Figure 1. This flexibility is particularly important in already built assembly lines (re-balancing context), which does not present many possibilities to reallocate tasks due to costly changes during the production phase. The model brings advantages to re-balancing problems or balancing problems with very restrictive precedence relations.

Both the assembly line balancing problem and the traveling salesman problem are known to be NP-hard (Baker, 1974; Papadimitriou, 1977). Instances combining those two problems can be much more difficult to solve than a simple assembly line balancing problem. The preprocessing procedures detailed in Section 4 are important to evaluate and reduce the number of variables that are associated to unfeasible or dominated solutions. The more restricted the instance is, the greater the reduction offered by the preprocessing, as indicated in Table 7 from three real cases tested.

The extended model (E-TWALBP) contemplates characteristics of real-world balancing lines that are not treated by simple assembly line balancing models. The re-balancing aspect is dealt with assignment restrictions in the allocation of tasks or workers. The model is also suitable to treat aspects of robotic and mixed labor assembly lines, as it is shown in Case Studies 1 and 3. Furthermore, common tasks (which require two or more workers) are implemented. Side workers or helpers are also assigned for common tasks and their movement time between stations is accounted for in the cycle time. Automatic tasks (which require a worker's trigger, but are performed by a machine) are also modeled and their impact considered as both workers and stations bind the cycle time.

Adapted literature datasets were employed to verify the model's performance with larger instances and to illustrate some of its features. The results of Section 5.1 showed that even without assignment restrictions (SALBP) there are improvement potential of considering extra stations in relation to the number of workers. Thus, the movement of workers acts as an alternative degree of freedom to obtain a better balancing (e.g., Tables 1-3). Although the larger and harder literature problems could not be solved to optimality, the real-world based instances could and are discussed in the case studies.

The presented case studies show the model potential and flexibility to treat real-world instances. The cases pose challenges such as human and robotic workers in the same line, automatic and common tasks, and the necessity to have workers to be assigned to more than one workstation (furthermore the movement time had to be taken into account). To the best of our knowledge, the union of all such characteristics are not yet found in literature datasets, which most frequently focus on SALBP and SALBP-instance-based variations. Improvements of 11.3%, 9.4%, and 12.7% in cycle times (Tables B.2, B.4, and B.7, respectively) were obtained in the study cases, verifying the practical utility of the model.

The model's hypothesis are presented (Section 2.2) and discussed (Section 6) based on the case studies. Although the verified limitations (linked to possible sequencing difficulties tied to common and automatic tasks) did not pose significant difficulties to the case studies, further works should address them and aim at a simultaneous balancing-scheduling approach for single model instances and balancing-sequencing for mixed-model instances.

Acknowledgment

We would like to acknowledge the financial support from Fundação Araucária (Agreement 141/2015) and CNPq (Grant 305405/2012-8).

Appendix A. Problem data

In this section, the data of the case studies are given. Table A.1 contains the basic parameters for each case such as number of tasks, workers, stations, and the production mixes. For the first case study, tasks' properties are given at Tables A.2 and A.3. The duration time of each task is shown for the two models and either for manual and robotic labor, but only model 1 is considered in the case study. The precedence diagram is described along with the assignment restrictions. Table A.4 contains the description of automatic and common tasks for the case 1, along with the assignment of the robot. The distance between each station is given by Table A.5. Note that the station 15 is reserved for the robot, and therefore no distance between the other stations is defined.

For Case Study 2, tasks parameters are shown in Table A.6. The table contains the duration time of each task in each machine, along with precedence relations and assignment restrictions. Table A.7 describes each workstation in the problem in terms of pieces produced per cycle, the pre-load applied to them and the zoning restriction (each machine is considered to be a worker). Once machines are fixed, no distances have to be defined.

Tables A.8 and A.9 contain the tasks' properties for case 3. The duration time, precedence relations, assignment restrictions and automatic tasks are listed in the tables. The robots' positions and the tasks they are able to perform are described in Table A.10. Note that robots can only perform one task each. These tasks contain all the operations the robots are responsible for, merged into a single operation. They perform specific tasks that cannot be done by human workers and once the cycle time of the robots is close to the required by the line, robot workloads are considered fixed. $TWInc$ must unable robots to perform any other tasks. Finally, Table A.11 contains the distance between each station.

Table A.1: Basic parameters for the case studies. Case 3 is run with a range of workers from 17 to 20. The *%Model* columns show the model's mix, however, the case studies considered only one model (Model 1).

Case	No. of Tasks	No. of Workers	No. of Stations	No. of Models	%Model 1	%Model 2
1	81	10	15	2	85%	15%
2	36	3	3	1	100%	
3	121	17 - 20	23	1	100%	

Table A.2: Task properties part 1 (Tasks 1 - 60) of Case Study 1. The *Duration Time* is the time needed to perform a task. Humans and robots have different processing speeds. Task's duration are given for both product models, but only model 1 is considered in the case study. The robot can only perform tasks that have entries for the cycle time. *Precedes Task* corresponds to the precedence diagram. This column indicates which task depends on the given task. The *Restricted to Stations* column states the assignment restrictions. A task can only be allocated to a station listed in this column.

	Human Duration Time		Robotic Duration Time			Restricted to
Task	Model 1	Model 2	Model 1	Model 2	Precedes Task	Stations
1	27	27			2	1
2	7	7			3, 4	1
3	11	11	8	8		1, 12, 15
4	2.5	2.5			46	1
5	68.5	68.5			13	2
6	68.6	68.6			13	3
7	3.5	0			8	4
8	12	0			9, 10	4
9	12	0	10	0		4, 7, 12, 15
10	21	15.2			11, 12	4
11	25	25				4
12	21	21				4
13	37.3	37.3			14, 15	5
14	24	18			16, 17	5
15	26	20	22	17.2		5, 7, 12, 15
16	21	15.5			18, 19	5
17	10	5	8.4	4.2		5, 7, 12, 15
18	16	16			20	5
19	5	5	5	5		5, 7, 12, 15
20	24.9	24.9			24	5
21	127.5	94.8			22, 23	6
22	21.3	21.3	18	18		6, 7, 12, 13
23	8	8			24	6
24	8.1	8.1			25	7
25	8	8			26	7
26	82.7	75			27, 28	7
27	9	9	8.1	8.1		7, 12, 15
28	2.6	2.6			29, 30	7
29	19	19			31	7
30	22	22	19.5	19.5		7, 12, 15
31	4.2	4.2			32, 33	7
32	2.8	2.8				7, 12
33	6	6			46	7
34	156.5	139.1			35, 36	8
35	30.2	24.9	27	22.3		8, 12, 13, 15
36	13.3	13.3			46	8
37	156.5	139.1			38, 39	9
38	30.2	24.9				9, 12, 13, 15
39	13.3	13.3			46	9
40	62.2	62.2			41	10
41	52.4	52.4			42, 43, 44	11
42	34	34			45	11
43	28	28	24	24		11, 12, 15
44	63	63	37	37		11, 12, 15
45	26.2	26.2			46	11
46	8.8	8.8			47, 48, 49	12
47	11.3	11.3	9.7	9.7		12, 13, 15
48	141	141	89	89		12, 13, 15
49	25.5	25.5			50	12
50	3	3			51	12
51	3.7	3.7			52	12
52	4.5	4.5			53	12
53	64.1	64.1			54, 55	12
54	10	10			56, 57	12
55	7	7	7	7		12, 13, 15
56	14	14			58, 59	12
57	11	11	9.4	9.4		12, 13, 15
58	9	9			60	12
59	29.8	29.8	26.3	26.3	61	12, 14, 15
60	9.7	9.7				12

Table A.3: Task properties part 2 (Tasks 61 - 81) of Case Study 1. The *Duration Time* is the time needed to perform a task. Humans and robots have different processing speeds. Task's duration are given for both product models, but only model 1 is considered in the case study. The robot can only perform tasks that have entries for the cycle time. *Precedes Task* corresponds to the precedence diagram. This column indicates which task depends on the given task. The *Restricted to Stations* column states the assignment restrictions. A task can only be allocated to a station listed in this column.

	Human Duration Time		Robotic Duration Time			Restricted to
Task	Model 1	Model 2	Model 1	Model 2	Preceeds Task	Stations
61	18	13.9			62, 63	13
62	2	2	2	2		13, 15
63	7.5	7.5			64, 65	13
64	18	18			66	13
65	2	2	2	2		13, 15
66	10.5	10.5			67	13
67	1.7	1.7			68, 69	13
68	4	4			70	13
69	4	4	3.7	3.7		13, 15
70	7.5	7.5			71	13
71	4.2	4.2			72, 73, 74	13
72	11.2	11.2			75, 76, 77, 78 79, 80, 81	13, 14
73	54.6	54.6	39	39		13, 15
74	62	52	46.5	36.5		13, 15
75	109	109				14
76	10	5				14
77	16	8				14
78	23	23				14
79	40.5	40.5				14
80	9.2	9.2				14
81	4.8	4.8				14

Table A.4: Special features of Case Study 1. Automatic tasks are the ones that do not count to the worker cycle time. Common tasks are represented by the number of the task and the amount of workers needed in brackets. The amount of robot workers is shown in the *Robotic Workers* along with the station the robot is allocated in brackets.

Automatic Tasks	72
Common Tasks	25(2), 50(2), 52(2), 58(2)
Robotic Workers	1(15)

Table A.5: Station distance matrix for the Case Study 1. The temporal distances are measured in the same time units as the tasks.

Station	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0.3	0.3	1.755	2.55	4.8	3.6	1.875	4.575	4.05	3.6	3.15	3	3.15
2		0	0	1.65	2.55	4.35	3.3	1.575	4.2	3.75	3.3	2.85	2.7	2.85
3			0	1.65	2.55	4.35	3.3	1.575	4.2	3.75	3.3	2.85	2.7	2.85
4				0	2.85	4.05	3.75	1.125	4.65	2.7	2.4	2.225	1.95	1.2
5					0	4.05	1.125	1.5	1.95	3.15	2.85	1.35	2.1	4.35
6						0	3	2.7	2.325	1.2	1.05	1.575	1.2	3.9
7							0	2.25	0.75	1.5	1.65	0.45	3	4.95
8								0	3.15	2.025	1.575	1.125	1.05	2.7
9									0	1.35	1.65	1.125	2.25	6
10			Simetrical Matrix							0	0.3	0.75	0.75	4.35
11											0	0.75	0.75	4.35
12												0	0.6	3.75
13													0	3.6
14														0

Table A.6: Task properties of Case Study 2. The duration time is the time needed to perform a task and varies with the station. Tasks can only be allocated to stations in which the duration time is defined. *Precedes Task* corresponds to the precedence diagram. This column indicates which task depends on the given task.

	Duration time in each station			
Task	Station 1	Station 2	Station 3	Precedes Task
1	53.2			2
2	19.3			3
3	11.7			All Tasks $\geq$ 4
4	13.5	9	18.4	
5	29.4	18.3		
6	39.2	30.5	71.6	23
7	16.1			11
8	26.7			9, 10
9	26			
10	6.9			
11	15			
12	18			
13	6.4	4.1	8.8	
14		10	22.7	
15		17.8		17
16	7.9	6.2		
17		24.7		
18		21.2		19
19		29.2		
20			26.3	
21	9.2	6.5	15	
22	37.3	29.4	65.9	23
23			34.3	
24			42.9	
25	13	8.9	17	26
26			16.9	
27			22.8	
28			24.9	
29			32.7	
30			8.8	
31	38.2		63.8	
32			12.9	33
33			15.4	
34			20.2	35
35			33	36
36			26.2	

Table A.7: Station properties of Case Study 2. *Pieces produced per cycle* is the *PpS* factor used in the model. The *Pre-Load* is the time needed to change pallets and it is considered a fixed pre-load for each machine. Once every machine is considered a worker, they are fixed to one position, showed in the column *Assigned Worker*.

Station	Pieces produced per cycle	Pre-Load	Assigned Worker
1	2	3	1
2	1	3.4	2
3	4	3	3

Table A.8: Task properties part 1 (Tasks 1-60) for Case Study 3. The duration time of a task, precedence relations, and assignment restrictions are shown respectively in each column. Automatic tasks are described in *Restricted to Station* column.

Task	Duration Time	Precedes Task	Restricted to Station
1	150	2	1
2	1200	3	2
3	140	4	
4	638	5	3
5	73	7	3
6	60	7	
7	113	8	3
8	360	9	4
9	48	10	
10	103	54	4
11	85	13	
12	58	13	
13	35	16	
14	28	16	
15	40	16	
16	285	17	
17	510	18	5
18	60	19	
19	43	20	
20	175	21	6
21	73	22	
22	63	24	6 (automatic)
23	33	24	
24	75	25	6
25	23	26	6
26	63	28	6 (automatic)
27	25	28	6
28	330	29	6
29	95	30, 31, 49	6
30	65	100	
31	38	44	6
32	25	34	
33	113	34, 42	
34	175	35	
35	200	36	8
36	18	37	
37	100	38, 40, 41, 42, 67	8
38	38	39	
39	60	40	
40	35	54	
41	75	42	
42	175	43	
43	338	64	8
44	118	45	7
45	168	46	7
46	88	47	
47	68	48	7
48	35	51	7
49	48	50	
50	75	51	
51	50	52	7
52	300	53, 87	7
53	125	94, 97, 100	7
54	195	55, 56, 57	9
55	55	58	
56	95	58	
57	38	58	
58	120	59	9
59	85	60	
60	75	62	

Table A.9: Task properties part 2 (Tasks 61-121) for Case Study 3. The duration time of a task, precedence relations, and assignment restrictions are shown respectively in each column. Automatic tasks are described in *Restricted to Station* column.

Task	Duration Time	Precedes Task	Restricted to Station
61	30	62	8, 9
62	98	63	
63	1247	64	10
64	280	65	
65	508	66	11
66	55	68	
67	73	68	8, 11
68	25	69	
69	113	70	
70	168	71	
71	95	72	
72	93	73	
73	68	74	
74	108	75	
75	95	76	
76	310	77	
77	165	79	
78	83	79	
79	278	80	
80	173	81	
81	75	82	
82	498	83	14
83	118	84	
84	118	85	
85	400	86	
86	425	87	
87	200	88	15
88	150	89	
89	125	90	
90	143	91	15
91	1312	92, 96, 107, 108	16
92	70	93	
93	25	102	
94	18	95	
95	25	102	
96	40	98	17
97	18	98	
98	140	99	
99	545	101, 102	17
100	90	101	
101	75	102	
102	13	103, 106	17
103	133	104	
104	860	105	18 (automatic)
105	23	109	
106	83	109	
107	40	109	
108	90	109	
109	120	110	
110	400	111	
111	75	112	
112	63	113	
113	680	114	19
114	858	115	20
115	428	116	21
116	325	118	22
117	40	118	22
118	650	119	22
119	250	121	22
120	80	121	22
121	555		23

Table A.10: Position of robots and tasks they perform in Case Study 3.

Robot	Worker Number	Fixed to Station	Able to Perform Task
1	9	10	63
2	15	16	91

Table A.11: Distance matrix between workstations for the Case Study 3. The matrix presents symmetrical distances.

Station	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.7	2.8	5.6	9.8	15.4	15.4	19.6	20.79	18.83	16.94	15.05
2		0	2.8	5.6	9.8	15.4	15.4	19.6	20.79	18.83	16.94	15.05
3			0	2.8	7	12.6	12.6	16.8	18.2	16.24	14.42	12.6
4				0	4.2	9.8	9.8	14	15.61	13.79	12.04	10.43
5					0	5.6	5.6	9.8	11.9	10.5	8.96	7.84
6						0	0.7	4.2	8.19	7.35	7	7.35
7							0	4.2	8.19	7.35	7	7.35
8								0	7	7.28	8.16	9.38
9									0	7.28	8.19	9.45
10										0	7.28	8.19
11											0	7.28
12												0

Station	13	14	15	16	17	18	19	20	21	22	23
1	13.23	9.87	8.54	7.56	7	7.14	7.84	8.96	10.43	11.2	22.4
2	13.23	9.87	8.54	7.56	7	7.14	7.84	8.96	10.43	11.2	22.4
3	10.5	8.19	7.28	7	7.28	8.19	9.38	10.5	12.6	14.42	25.62
4	8.96	7.14	7	7.56	8.54	9.87	11.48	13.23	15.05	16.94	28.14
5	7.14	8.54	9.87	11.48	13.23	15.05	16.94	18.9	20.79	22.82	34.02
6	8.19	12.6	14.42	16.24	18.2	20.16	22.12	24.15	24.5	26.18	37.38
7	8.19	12.6	14.42	16.24	18.2	20.16	22.12	24.15	24.5	26.18	37.38
8	10.92	14.42	16.24	18.2	20.16	22.12	24.15	26.18	27.3	28	39.2
9	10.92	14.42	16.31	18.2	20.16	22.12	24.15	26.18	27.3	28	39.2
10	9.45	12.6	14.42	16.31	18.2	20.16	22.12	24.15	26.18	27.3	38.5
11	8.19	10.92	12.6	14.42	16.31	18.2	20.16	22.12	24.15	24.5	35.7
12	7.28	9.45	10.92	12.6	14.42	16.31	18.2	20.16	22.12	22.4	33.6
13	0	8.19	9.45	10.92	12.6	14.42	16.31	18.2	20.16	21	32.2
14		0	8.19	9.45	10.92	12.6	14.42	16.31	18.2	19.6	30.8
15			0	8.19	9.45	10.92	12.6	14.42	16.31	17.5	28.7
16				0	8.19	9.45	10.92	12.6	14.42	15.4	26.6
17					0	8.19	9.45	10.92	12.6	14	25.2
18						0	8.19	9.45	10.92	11.9	23.1
19							0	8.19	9.45	10.5	21.7
20								0	8.19	9.1	20.3
21									0	8.4	19.6
22										0	21
23											0

Appendix B. Detailed results

This section contains the answers obtained by the model. The most relevant answers are the tuples containing the assigned TWS and WSS values. With these values, we can completely define all allocations and the routing cycle of each worker. Obtained cycle time for each workstation and each worker, along with his/her respective movement time, are also shown in this section.

The assignment obtained answers (TWS values) are represented in Tables B.1, B.3, and B.5 for the Cases 1, 2, and 3 respectively. The movements performed by each worker are described with the WSS values in Tables B.6 for Case Study 3 (and Table 6 for Case Study 1). Case 2 does not contain movements. Finally, Tables B.2, B.4, and B.7 account for the cycle time for each worker and station along with the movement time of each worker.

Table B.1: Task, worker, and station assignments for the Case Study 1. Tasks 25, 50, 52, and 58 are common tasks and task 72 is an automatic task (Table A.4). Note that common tasks have two-worker assignments.

T	W	S	T	W	S	T	W	S	T	W	S
1	1	1	23	2	6	43	7	11	61	3	13
2	1	1	24	4	7	44	10	15	62	10	15
3	1	1	25	4	7	45	7	11	63	3	13
4	1	1	25	6	7	46	8	12	64	3	13
5	1	2	26	4	7	47	8	12	65	3	13
6	2	3	27	4	7	48	10	15	66	3	13
7	1	4	28	4	7	49	8	12	67	3	13
8	1	4	29	4	7	50	6	12	68	3	13
9	1	4	30	4	7	50	8	12	69	3	13
10	1	4	31	4	7	51	8	12	70	3	13
11	1	4	32	8	12	52	7	12	71	3	13
12	1	4	33	4	7	52	8	12	72	3	13
13	3	5	34	5	8	53	8	12	73	10	15
14	3	5	35	5	8	54	8	12	74	10	15
15	4	7	36	5	8	55	8	12	75	9	14
16	3	5	37	6	9	56	8	12	76	9	14
17	3	5	38	6	9	57	8	12	77	9	14
18	3	5	39	6	9	58	5	12	78	9	14
19	4	7	40	7	10	58	8	12	79	9	14
20	3	5	41	7	11	59	8	12	80	9	14
21	2	6	42	7	11	60	8	12	81	9	14
22	4	7									

Table B.2: Resulting cycle time for each workstation and each worker in Case Study 1. The movement time for each worker is also described. Note that the cycle time is restricted by worker 3 (211.78 TU). The original configuration was limited at 240.49 TU (Worker 6).

	Model			Original		
Stations	Station CT	Worker CT	Movement	Station CT	Worker CT	Movement
1	47.5	209.21	3.705	47.5	233.058	3.705
2	68.5	209.425	5.1	68.5	160.825	0
3	68.6	211.78	2.4	68.6	204.828	8.18
4	89.505	209.645	0	94.5	171.245	0
5	135.725	207.845	2.25	164.2	238.088	2.1
6	130.595	209.92	2.325	156.8	240.488	4.5
7	209.645	209.1	0.6	164.4	221.1	1.8
8	196.595	211.4	0	200	211.4	0
9	196.595	210.55	0	200	210.55	0
10	62.2	210	0	62.2	210	0
11	140.6			140.6		
12	211.4			211.4		
13	89.985			79.4		
14	210.55			223.7		
15	210			210		

Table B.3: Task, worker, station assignments for Case Study 2. The original assignment is also displayed in the right part of the table.

Model Result Assignment						Original Assignment					
T	W	S	T	W	S	T	W	S	T	W	S
1	1	1	19	2	2	1	1	1	19	2	2
2	1	1	20	3	3	2	1	1	20	3	3
3	1	1	21	1	1	3	1	1	21	3	3
4	3	3	22	1	1	4	1	1	22	3	3
5	2	2	23	3	3	5	1	1	23	3	3
6	3	3	24	3	3	6	1	1	24	3	3
7	1	1	25	3	3	7	1	1	25	3	3
8	1	1	26	3	3	8	1	1	26	3	3
9	1	1	27	3	3	9	1	1	27	3	3
10	1	1	28	3	3	10	1	1	28	3	3
11	1	1	29	3	3	11	1	1	29	3	3
12	1	1	30	3	3	12	1	1	30	3	3
13	3	3	31	3	3	13	2	2	31	3	3
14	2	2	32	3	3	14	2	2	32	3	3
15	2	2	33	3	3	15	2	2	33	3	3
16	1	1	34	3	3	16	2	2	34	3	3
17	2	2	35	3	3	17	2	2	35	3	3
18	2	2	36	3	3	18	2	2	36	3	3

Table B.4: Resulting cycle time for each workstation for the optimization model and the original assignment applied to the machines (Case Study 2). The bottleneck station moved from workstation 1 (140.5 TU) in the original assignment to workstation 3 (127.225 TU) in the model's response.

Worker	Model Answer	Original
1	126.65	140.5
2	124.6	116.6
3	127.225	122.75

Table B.5: Task, worker, and station (TWS) assignments for the Study Case 3.

T	W	S	T	W	S	T	W	S	T	W	S
1	1	1	32	5	6	62	4	9	92	13	17
2	2	2	33	6	7	63	9	10	93	13	17
3	2	2	34	6	7	64	8	11	94	6	7
4	3	3	35	7	8	65	8	11	95	11	14
5	3	3	36	7	8	66	8	11	96	13	17
6	1	1	37	7	8	67	7	8	97	7	8
7	3	3	38	7	8	68	8	11	98	13	17
8	3	4	39	7	8	69	8	11	99	13	17
9	3	4	40	7	8	70	8	11	100	13	17
10	3	4	41	7	8	71	8	11	101	13	17
11	1	1	42	7	8	72	8	11	102	13	17
12	1	1	43	7	8	73	10	12	103	13	17
13	1	1	44	6	7	74	10	12	104	14	18
14	1	1	45	6	7	75	10	12	105	14	18
15	1	1	46	6	7	76	10	12	106	13	17
16	1	1	47	6	7	77	10	12	107	13	17
17	4	5	48	6	7	78	6	7	108	13	17
18	5	6	49	5	6	79	10	12	109	16	19
19	5	6	50	5	6	80	10	12	110	16	19
20	5	6	51	6	7	81	11	14	111	16	19
21	5	6	52	6	7	82	11	14	112	16	19
22	5	6	53	6	7	83	11	14	113	16	19
23	5	6	54	4	9	84	11	14	114	14	20
24	5	6	55	4	9	85	11	14	115	14	21
25	5	6	56	4	9	86	12	15	116	17	22
26	5	6	57	4	9	87	12	15	117	17	22
27	5	6	58	4	9	88	12	15	118	17	22
28	5	6	59	4	9	89	12	15	119	17	22
29	5	6	60	4	9	90	12	15	120	17	22
30	7	8	61	7	8	91	15	16	121	1	23
31	5	6									

Table B.6: Movements performed by the workers in Study Case 3. A Worker W moves from station S_p to station S_s.

W	S_p	S_s
1	1	23
1	23	1
2	2	2
3	3	4
3	4	3
4	5	9
4	9	5
5	6	6
6	7	7
7	8	8
8	11	11
9	10	10
10	12	12
11	14	14
12	15	15
13	17	17
14	18	20
14	20	21
14	21	18
15	16	16
16	19	19
17	22	22

Table B.7: Resulting cycle time for each workstation and each worker in Case Study 3. The movement time for each worker is also described. Note that the cycle time is restricted by worker 17 (1345 TU). The total movement time accounts for 102.76 TU. The original cycle time was limited by Worker 2 (1540.6 TU).

	Model			Original	
Stations	Station CT	Worker CT	Movement	Worker CT	Movement
1	741	1340.8	44.8	1351.4	1.4
2	1340	1340	0	1540.6	5.6
3	824	1340.6	5.6	1101	0
4	511	1294.8	23.8	1101	0
5	510	1118	0	1352	0
6	1244	1341	0	949	0
7	1341	1225	0	791	0
8	1225	1337	0	941	0
9	761	1247	0	1247	0
10	1247	1197	0	740	0
11	1337	1234	0	1009	0
12	1197	1043	0	1209	0
13	0	1344	0	1043	0
14	1234	1337.56	28.56	1312	0
15	1043	1312	0	1059	0
16	1312	1338	0	1027	0
17	1344	1345	0	680	0
18	883			1302.38	16.38
19	1338			1225	0
20	858			675	0
21	428				
22	1345				
23	555				

References

Altemeier, S., Helmdach, M., Koberstein, A., & Dangelmaier, W. (2010). Reconfiguration of assembly lines under the influence of high product variety in the automotive industry–a decision support system. *International Journal of Production Research*, *48*, 6235–6256. doi:10.1080/00207540903252290.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*. New York: Wiley.

Battaïa, O., Delorme, X., Dolgui, A., Hagemann, J., Horlemann, A., Kovalev, S., & Malyutin, S. (2015). Workforce minimization for a mixed-model assembly line in the automotive industry. *International Journal of Production Economics*, *170*, 489–500. doi:10.1016/j.ijpe.2015.05.038.

Battaïa, O., & Dolgui, A. (2012). Reduction approaches for a generalized line balancing problem. *Computers & Operations Research*, *39*, 2337–2345. doi:10.1016/j.cor.2011.11.022.

Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, *177*, 2016–2032. doi:10.1016/j.ejor.2005.12.017.

Baybars, I. (1986). Survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, *32*, 909–932.

Borba, L., & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the Assembly Line Worker Assignment and Balancing Problem. *Computers & Operations Research*, *45*, 87–96. doi:10.1016/j.cor.2013.12.002.

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, *183*, 674–693. doi:10.1016/j.ejor.2006.10.010.

Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics*, *111*, 509–528. doi:10.1016/j.ijpe.2007.02.026.

Costa, A. M., & Miralles, C. (2009). Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics*, *120*, 625–632. doi:10.1016/j.ijpe.2009.04.013.

Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). Solutions of a large scale travelling salesman problem. *Operations Research*, *2*, 393–410.

Falkenauer, E. (2005). Line balancing in the real world. In *Proceedings of the International Conference on Product Lifecycle Management, PLM 05*. Lyon.

Fattahi, P., Roshani, A., & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, *53*, 363–378. doi:10.1007/s00170-010-2832-y.

Gamberini, R., Gebennini, E., Grassi, A., & Regattieri, A. (2009). A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem. *International Journal of Production Research*, *47*, 2141–2164. doi:10.1080/00207540802176046.

Gronalt, M., & Hartl, R. F. (2003). Workforce planning and allocation for mid-volume truck manufacturing: A case study. *International Journal of Production Research*, *41*, 449–463. doi:10.1080/00207540210162974.

Gujjula, R., & Gunther, H.-O. (2009). Scheduling utility workers at mixed-model assembly lines. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1092–1096). IEEE. doi:10.1109/IEEM.2009.5372954.

Hamzadayi, A., & Yildiz, G. (2013). A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines. *Computers & Industrial Engineering*, *66*, 1070–1084. URL: http://www.sciencedirect.com/science/article/pii/S0360835213002623. doi:10.1016/j.cie.2013.08.008.

Lopes, T. C., Sikora, C. G. S., & Magatão, L. (2016). Buffer and Cyclical Product Sequence Aware Assembly Line Balancing Problem: Model and Steady-State Balancing Case Study. In *XLVIII Simpósio Brasileiro de Pesquisa Operacional (XLVIII Brazilian Symposium of Operational Research)* (pp. 1–12).

Makssoud, F., Battaïa, O., Dolgui, A., Mpofu, K., & Olabanji, O. (2015). Re-balancing problem for assembly lines: new mathematical model and exact solution method. *Assembly Automation*, *35*, 16–21.

Mayrhofer, W., März, L., & Sihn, W. (2013). Simulation-based optimization of personnel assignment planning in sequenced commercial vehicle assembly: A software tool for iterative improvement. *Journal of Manufacturing Systems*, *32*, 423–428. doi:10.1016/j.jmsy.2013.04.009.

Miller, C. E., Tucker, A. W., & Zemlin, E. A. (1960). Integer programming formulation of travelling salesman problems. *Journal of the ACM*, *3*, 326–329.

Miltenburg, J. (1998). Balancing U-lines in a multiple U-line facility. *European Journal of Operational Research*, *109*, 1–23.

Miltenburg, J. (2001). One-piece flow manufacturing on U-shaped production lines: a tutorial. *IIE Transactions*, *33*, 303–321. doi:10.1080/07408170108936831.

Miltenburg, J., & Wijngaard, J. (1994). U-line line balancing problem. *Management Science*, *40*, 1378–1388.

Miralles, C., García-Sabater, J. P., Andrés, C., & Cardós, M. (2008). Branch and bound procedures for solving the Assembly Line Worker Assignment and Balancing Problem: Application to Sheltered Work centres for Disabled. *Discrete Applied Mathematics*, *156*, 352–367. doi:10.1016/j.dam.2005.12.012.

Moreira, M. C. O., Miralles, C., & Costa, A. M. (2015). Model and heuristics for the Assembly Line Worker Integration and Balancing Problem. *Computers & Operations Research*, *54*, 64–73. doi:10.1016/j.cor.2014.08.021.

Nakade, K., & Nishiwaki, R. (2008). Optimal allocation of heterogeneous workers in a U-shaped production line. *Computers & Industrial Engineering*, *54*, 432–440. doi:10.1016/j.cie.2007.08.007.

Nakade, K., & Ohno, K. (2003). Separate and carousel type allocations of workers in a U-shaped production line. *European Journal of Operational Research*, *145*, 403–424. doi:10.1016/S0377-2217(02)00157-1.

Otto, A., Otto, C., & Scholl, A. (2013). Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. *European Journal of Operational Research*, *228*, 33–45. doi:10.1016/j.ejor.2012.12.029.

Öztürk, C., Tunali, S., & Hnich, B. (2013). Balancing and scheduling of flexible mixed model assembly lines. *Constrains*, *18*, 434–469.

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

Papadimitriou, C. H. (1977). The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, *4*, 237–244.

Patterson, J. H., & Albracht, J. J. (1975). Assembly-Line Balancing: Zero-One Programming with Fibonacci Search. *Operations Research*, *23*, 166–172. doi:10.1287/opre.23.1.166.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines*. (2nd ed.). Heidelberg: Physica.

Scholl, A., & 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. doi:10.1016/j.ejor.2004.07.022.

Scholl, A., Fliedner, M., & Boysen, N. (2010). Absalom: Balancing assembly lines with assignment restrictions. *European Journal of Operational Research*, *200*, 688–701. doi:10.1016/j.ejor.2009.01.049.

Shewchuk, J. P. (2008). Worker allocation in lean U-shaped production lines. *International Journal of Production Research*, *46*, 3485–3502. doi:10.1080/00207540601115997.

Sikora, C. G. S., Lopes, T. C., & Magatão, L. (2015). Variable Sets Reduction for Assembly Line Balancing Problem: MILP Model and Case Studies. In *XLVII Simpósio Brasileiro de Pesquisa Operacional (XLVIII Brazilian Symposium of Operational Research)* (pp. 166–176).

Sirovetnukul, R., & Chutima, P. (2010). The Impact of Walking Time on U-Shaped Assembly Line Worker Allocation Problems. *Engineering Journal*, *14*, 53–78.

Sparling, D., & Miltenburg, J. (1998). The mixed-model U-line balancing problem. *International Journal of Production Research*, *36*, 485–501.

Sternatz, J. (2014). Enhanced multi-Hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotive industry. *European Journal of Operational Research*, *235*, 740–754.

Sungur, B., & Yavuz, Y. (2015). Assembly line balancing with hierarchical worker assignment. *Journal of Manufacturing Systems*, *37*, 290–298. doi:10.1016/j.jmsy.2014.08.004.

Thomopoulos, N. T. (1970). Mixed Model Line Balancing with Smoothed Station Assignments. *Management Science*, *16*, 593–603. doi:10.1287/mnsc.16.9.593.

Vilà, M., & Pereira, J. (2014). A branch-and-bound algorithm for assembly line worker assignment and balancing problems. *Computers & Operations Research*, *44*, 105–114. doi:10.1016/j.cor.2013.10.016.

Yazgan, H. R., Beypinar, I., Boran, S., & Ocak, C. (2011). A new algorithm and multi-response Taguchi method to solve line balancing problem in an automotive industry. *The International Journal of Advanced Manufacturing Technology*, *57*, 379–392. doi:10.1007/s00170-011-3291-9.

Zha, J., & Yu, J.-j. (2014). A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment. *Journal of Manufacturing Systems*, *33*, 93–102. doi:10.1016/j.jmsy.2013.08.002.