Benders' decomposition for the balancing of assembly lines with stochastic demand

Celso Gustavo Stall Sikora*

Moorweidenstr. 18, 20148, Hamburg, Institute for Operations Research, University of Hamburg

Abstract

The quality of the balancing of mixed-model assembly lines is intimately related to the defined production sequence. The two problems are, however, incompatible in time, as balancing takes place when planning the line, while sequencing is an operational problem closely related to market demand fluctuations. In this paper, an exact procedure to solve the integrated balancing and sequencing problem with stochastic demand is presented. The searched balancing solution must be flexible enough to cope with different demand scenarios. A paced assembly line is considered and utility work is used as a recourse for station border violations. A Benders' decomposition algorithm is developed along with valid inequalities and preprocessing as a solution procedure. Three datasets are proposed and used to test algorithm performance and the value of treating uncertainty in mixed-model assembly lines. The integration of the strategic balancing problem with the operational sequencing problem results in more robust assembly lines.

Keywords: flexible manufacturing systems, assembly line balancing problem, stochastic optimization, combinatorial Benders' decomposition, mixed-model assembly line

DOI: 10.1016/j.ejor.2020.10.019

*Corresponding author
Email address: celso.sikora@uni-hamburg.de (Celso Gustavo Stall Sikora)

1. Introduction

Assembly lines are highly specialized production arrangements suitable for a high volume of homogeneous goods. The production activities are split between several workstations, each one performing elementary tasks. With this flow-oriented division of the process, production levels of, for instance, one car per minute are easily achievable nowadays (Emde & Gendreau, 2017). The market demand is, however, not suitable for the mass production of a single product. The color, engine power, in-board technologies, or comfort upgrades are examples of a large set of customizations a customer can choose in the automotive market. The production under such variety is called mass customization (Boysen et al., 2008). The planning of assembly lines faces a dilemma: they have to be flexible to allow the production of customized goods and, at the same time, they must be highly specialized to achieve high production levels.

The efficient use of assembly lines depends on a set of short and long term decisions. The implementation of the assembly line itself with all machinery and tools can be seen as a medium to long-lasting investment (Boysen et al., 2007). The decision of how to divide the process between workstations is called Assembly Line Balancing Problem (ALBP), for which the efficiency is based on how well distributed the processing time is. The production planning may also have to consider more factors, such as, just to cite a few, equipment selection (Oesterle et al., 2017), whether to use robotic or manual work, how many workplaces to build at each station (Michels et al., 2018), and the space for the pieces and tools (Chica et al., 2018). At the operational level, related short-term decisions can also greatly influence the efficiency of a line. Examples are the sequencing of models to be produced based on the realized or projected sales, the routing and scheduling of tow trains based on the weekly demand (Emde & Boysen, 2012), and the assignment of workers in case of absence (Ritt et al., 2016) or job rotation (Mossa et al., 2016).

In this paper, the combination of assembly line balancing and product mix sequencing is tackled. Both problems present different time scales, as the planning of an assembly line is a several months to years decision, while the sequencing may be determined daily or weekly. It follows that, during the lifespan of an assembly line, the demand may fluctuate and several different sequences might be used. A combined optimization approach is more meaningful when multiple demand scenarios are considered.

The balancing of mixed-model assembly lines with uncertain demand has been treated in the context of robust optimization and the Make-to-Order philosophy (see literature review in Sec. 2). Differently, this paper proposes a stochastic view of the balancing and sequencing problem: the approach considers demand forecasts and aims at the best expected balancing on average for its lifespan. The balancing problem is solved in combination with multiple potential short-term sequencing scenarios. The sequences are considered to be cyclic, that is, the models are repeated in the optimal sequence throughout production.

In this work, the balancing and cyclical sequencing of paced mixed-model assembly lines is considered in a stochastic two-stage framework, with balancing as its first stage and sequencing as the second stage. The expected utility work necessary for operating the assembly line is computed using Benders' decomposition. As the second-stage (sequencing) contains binary variables, combinatorial benders' cuts and valid

inequalities are proposed. Furthermore, a new approach for generating cuts based on the decomposition of the second stage is proposed, the partial combinatorial cuts. It is shown that lower bounds based on the further decomposition of the second stage can produce useful combinatorial cuts.

The remaining sections of the paper are structured as follows. A literature review on assembly line balancing under uncertainty is presented in Section 2, focusing on the uncertain production demand. The problem definition and a mathematical formulation are introduced in Section 3, while the Benders' decomposition procedure is described in Section 4. More specifically, the new partial combinatorial cuts are introduced in Section 4.4. Computational studies based on three proposed datasets are presented in Section 5, followed by a critique of the approach and suggestions for further work in the conclusion (Section 6).

2. Literature review on balancing under uncertain demand

In this section, the literature review on demand uncertainty in balancing of assembly lines is presented. Although balancing problems can present themselves in several variations, the section focuses mostly on serial lines with base characteristics. Therefore, papers about related problems such as buffer allocation or balancing varieties such as parallel-stations, U-shaped lines, two-sided lines, and multi-manned stations are only mentioned in relation to their stochastic characteristics. Extensive reviews on assembly line problem variations are surveyed by Boysen et al. (2007, 2008) and Battaïa & Dolgui (2013).

By far the most references on the stochasticity in assembly lines deal with uncertain processing times. Nondeterministic processing times effect the calculation of the workload of a station, which is defined as a random variable. This uncertainty causes complications on the balancing decisions because the total workload can surpass the aimed cycle time depending on the realization of random processing times.

For the cases in which the processing times exceed the planned cycle time, a remedial action is used to cope with the variability. Appropriate remedial actions can be the stoppage of the line, utility work, off-line repair, or higher production speed at the cost of product quality (Boysen et al., 2008). An alternative is the use of unpaced lines, for which the movement of pieces is discontinuous and occurs only when the processing is completed. In this case, the cycle time itself is a random variable.

The literature on stochastic balancing is rich and is the focus of a survey by Bentaha et al. (2015). Although several modeling approaches exist, most of the works can be summarized in three categories with respect to their premises and objectives. The first modeling alternative is to consider the sizing cost as given and minimize the stochastic corrective cost for a given system structure. As an example, Reeve & Thomas (1973) consider an assembly line with a fixed number of stations and a given cycle time. The authors search for the task assignment with the lowest probability of surpassing the time threshold. Alternatively, as a second approach, the system sizing cost is minimized for a necessary minimal system availability. In this direction, Kao (1976) defines a cost minimization assembly line balancing variant for a minimal degree of reliability on the completion of tasks. In other words, the approach minimizes the

number of stations under the condition that $Pr(WL_s \leq CT) \geq \alpha$ for all stations s, that is, the probability that the workload of each station s (WL_s) is smaller than the planned cycle time (CT) must be larger than a predefined threshold α (usually 90% or 95%). Sphicas & Silverman (1976) showed that for some distribution functions of the processing time, the stochastic problem based on $Pr(WL_s \leq CT) \geq \alpha$ can be translated to a deterministic version in which $E(WL_s) \leq b \cdot CT$ for some $0 < b \leq 1$.

While the first and second approaches fix either one of the sizing or reliability costs, a third approach considers the economic effects of the reliability, minimizing the sum of both costs. The pricing of the unreliability depends on the remedial action applied in the system. One example is given by Kottas & Lau (1973), who consider a trade-off between the number of workers on the assembly line and the costs of correcting the not completed tasks at the end of the line. Silverman & Carter (1986) stop the line if a task is not finished until the exceeded operation is concluded. They minimize the line cost, considering both the number of the stations and the probability of line stoppage, translated into production rate. Differently, unpaced lines have variable cycle times as the pieces only advance when all its operations are performed. Tiacci (2015a), for instance, uses simulation to approximate the performed processing times and focus on the maximization of line throughput.

Compared to the stochasticity of the processing time, few articles deal with uncertain product demand. Considering variations between product mixes requires the assumption of multiple products and, therefore, a mixed-model assembly line. In the next paragraphs, approaches that consider models individually and others that gather model information in an average model are discussed.

The first works on the stochastic balancing of mixed model assembly lines reduce the problem to a single model case. Just as in the seminal work of Thomopoulos (1967), an average model has its processing times as the weighted average of the production mix. In the context of balancing under uncertainty, Vrat & Virani (1976) consider multiple models as a single model with stochastic processing times. The different produced models are considered responsible for the variation and the problem is solved as a single model. McMullen & Frazier (1997) focus on multiple models with stochastic task durations. Their approach also simplifies to a single model problem. For each task, both the average and the variance on the processing times are aggregated. The reduction to a single model is also employed by Hop (2006) in the context of fuzzy processing times. These references are based on only one production mix.

More recently, publications in the context of flexible or robust assembly lines consider the variation of product mix in the form of scenarios. Simaria et al. (2009) study an assembly line in which the product or the volume of production varies. The authors consider production scenarios of single products and search for the balancing with the minimal amount of workstations to be used for all scenarios. Chica et al. (2013, 2016, 2018) consider scenario-based demand mixes in a robust optimization framework. They propose multi-objective formulations, in which the line sizing cost, station size, and solution robustness are regarded. These formulations are also based on the processing time of the average model or the most loaded model, that is, no sequencing is considered. Li & Gao (2014) model the scenarios with their

occurring probability and the balancing is solved based on the weighted average processing time. The authors consider the possibility of overtime to deal with highly loaded demand scenarios. The objective is to minimize the working costs, considering normal and overtime wages. Yang & Gao (2016) solves the balancing and rebalancing of assembly lines problems under stochastic demand. For different demand scenarios, the base balancing solution can be adapted by reassigning tasks to neighboring stations. The cycle time is calculated as the average of the models, that is, no sequencing is considered.

Modeling processing times as an average of multiple models can simplify the solution of the balancing problem. However, the production sequence of models plays a role in the efficiency of the line (Boysen et al., 2008). A solution based on the average processing time may violate the target cycle time without remedial actions that compensate for the time variation. Therefore, several authors have researched the effects of the sequences on assembly lines.

One trend in the literature is to solve the balancing for a random sequence of products. As the balancing and sequencing have different time frames, the balancing is solved considering that assembly line operates with the sequence of sold products. Authors call it a Make-to-Order condition and use the relative percentages of the product mix to generate random sequences to be produced. Bukchin et al. (2002) present a heuristic that uses a statistical bottleneck measure to evaluate solutions. They compute the difference between models and calculate, based on model probability frequencies, which stations would be the system bottleneck. The procedure works for unpaced assembly lines and aims at the minimization of the expected cycle time. Manavizadeh et al. (2012) extends the idea of Bukchin et al. (2002) with metaheuristics in a multi-objective context. Tiacci (2015a,b) similarly solves the balancing for random sequences of a product mix. Alternatively, the author uses simulation to evaluate the cycle time and a genetic algorithm to find balancing solutions.

A second approach explores the flexibility in the selection of product sequences. In industry, goods are often not produced exactly in the order they are purchased but scheduled in a daily, weekly, or even monthly plan (Boysen et al., 2009a). The mixed-model sequencing has been the focus of several authors, as showed in a survey by Boysen et al. (2009b). This problem is proven to be strongly NP-Hard (McCormick & Rao, 1994) for unpaced lines of three or more stations. For paced lines, Yano & Rachamadugu (1991) developed an analytical solution for the case of a single station and two models. For more stations, the authors use a heuristic. The approaches that combine balancing and sequencing are mostly deterministic. The solution literature consists of Karabati & Sayın (2003) for unpaced synchronous lines, Sawik (2012) and Öztürk et al. (2015) for unpaced asynchronous lines and Lopes et al. (2018) for both synchronous, asynchronous, or hybrid variants. In the stochastic context, Özcan et al. (2011) present a metaheuristic for the simultaneous balancing and sequencing under uncertain processing times. Additionally, Mosadegh et al. (2017) propose heuristics for the sequencing of models with uncertain processing times in the bottleneck station of a mixed model assembly line. An extension based on metaheuristics dealing with multiple stations is given by Mosadegh et al. (2019).

Table 1 contains a summary of the cited articles that deal with the stochastic balancing of mixed-model assembly lines. The references are classified with respect to problem scope, product variety, uncertain data, objective, and solution approach. For the objectives, a category "Others" is included. An example is an indirect objective, such as minimizing the difference or variance of processing times between models. Additionally, non-filled circles refer to partially fulfilling the classification. A non-filled circle for sequencing means that sequences are considered as random and therefore not optimized. Moreover, Simaria et al. (2009) consider assembly lines for multiple models, however, only one model is produced at a production shift. Chica et al. (2013) consider balancing, but do not propose a balancing solution.

Table 1: Literature overview for the stochastic balancing of mixed-model assembly lines.

	Problem scope		Product variety		Uncertain data			Objective				Solution method		
Author(s) (Year)	Balancing	Sequencing	Average model	Mixed-model	Processing time	Demand (probability)	Demand (scenarios)	Min Stochastic effect	Min Sizing cost	Min Combined cost	Others	Heuristic	Metaheuristic	Exact method
Vrat & Virani (1976)	•		•		•					•		•		
McMullen & Frazier (1997)	•		•		•				•			•		
Bukchin et al. (2002)	•	○		•		•			•			•		
Hop (2006)	•		•		•				•			•		
Simaria et al. (2009)	•			○			•	•					•	
Özcan et al. (2011)	•	•		•	•				•		•		•	
Manavizadeh et al. (2012)	•	○		•		•				•	•		•	
Chica et al. (2013)	○		•				•			•	•			
Li & Gao (2014)	•		•				•			•				•
Tiacci (2015a,b)	•	○		•	•	•				•			•	
Yang & Gao (2016)	•		•				•		•					•
Chica et al. (2016, 2018)	•		•				•			•	•		•	
Mosadegh et al. (2017)		•		•	•			•				•		•
Mosadegh et al. (2019)		•		•	•			•					•	•
Proposed paper (2021)	•	•		•			•	•						•

To the best of the author's knowledge, no article proposes the solution of the balancing along with the sequencing of stochastic demand scenarios. Although the stochastic demand for random sequences has been treated, the joint optimization of the sequence provides better results and is more consistent with what is applied in industry. Practitioners have the flexibility to schedule models in a daily or weekly time window. In contrast to the deterministic balancing approaches, the stochastic solution assures the task assignment is flexible enough to cope with a variety of possible demand scenarios efficiently. To fill out this gap, this paper presents a decomposition method to exactly solve the balancing and sequencing for a stochastic demand modeled in product mix scenarios. The cycle time and the number of the stations of a paced line are considered given, while the expected utility work is minimized.

3. Problem definition and mathematical formulation

The characteristics of the treated assembly line and the modeled uncertainty is presented in this section. Furthermore, a Mixed Integer Linear Programming (MILP) formulation is given.

In this paper, a paced assembly line is considered. Workstations are regarded to have fixed station boundaries that do not intercept with other stations. The conveyor length of each station is longer than necessary for the expected cycle time, so that flexibility is given for the processing of different models. That is, a less loaded model can potentially compensate more loaded models. Depending on the order in which pieces are produced, workers may not be able to finish the job within station bounds. If the work would extend beyond the defined station, a utility worker is assigned to finish the remaining operations before the piece arrives at the next station.

The sequencing of mixed-model assembly lines is frequently based on building blocks that can represent the demand. As a simplification strategy, Bard et al. (1992) define a minimum part set (MPS) as an efficient alternative to global sequencing. In the MPS, the smallest set containing the same proportion of the total demand is sequenced. In order to assure the relevance of the MPS in the sequencing of the total demand, cyclical scheduling is a very useful approach (McCormick et al., 1989) that is shown to converge quickly to a steady-state (Lopes et al., 2020). The cyclical approach based on MPS is used in the sequencing part of the proposed problem.

The scheduling of a mixed model paced assembly line is illustrated in Figure 1. Each bar corresponds to the processing of a workpiece ($P1-P4$). The velocity of the conveyor belt is defined as 1 length unit per time unit so that the distance is equivalent to time measurement. The cycle time (CT) is considered given and corresponds to the launch time difference of two adjacent pieces in a station. The vertical dashed line in Figure 1 represents the position of a piece when the next piece enters the station at position 0. The station length is conveniently measured based on a multiplier of the equivalent distance of the cycle time (length multiplier, LM).

In the example of Fig. 1, piece $P1$ requires more time than the cycle time so that the worker starts the processing of the next piece $P2$ after it has advanced some length of the line. If a piece is finished before the vertical dashed line (CT), the next piece is not yet available. In this case, idle time occurs until the next piece ($P3$) enters the station. In the example, the processing of the fourth piece ($P4$) requires the extension of the station limits. In this paper, such extension is avoided by employing utility work. As in the base model of the mixed model sequencing surveyed by Boysen et al. (2009b), the work overload is supposed to have no impact on succeeding stations. As a set of models is considered to be sequenced cyclically in the formulation, the starting position of the first piece of the next cycle ($P1$ - Cycle 2) must be the same as in the first cycle. Therefore, delays are also considered as violations, because, without utility work, the scheduling cannot repeat itself in a stable manner.

In this paper, the uncertainty based on the product mix is represented by a collection of demand scenarios. Each scenario contains a given product mix along with the probability of each mix to be

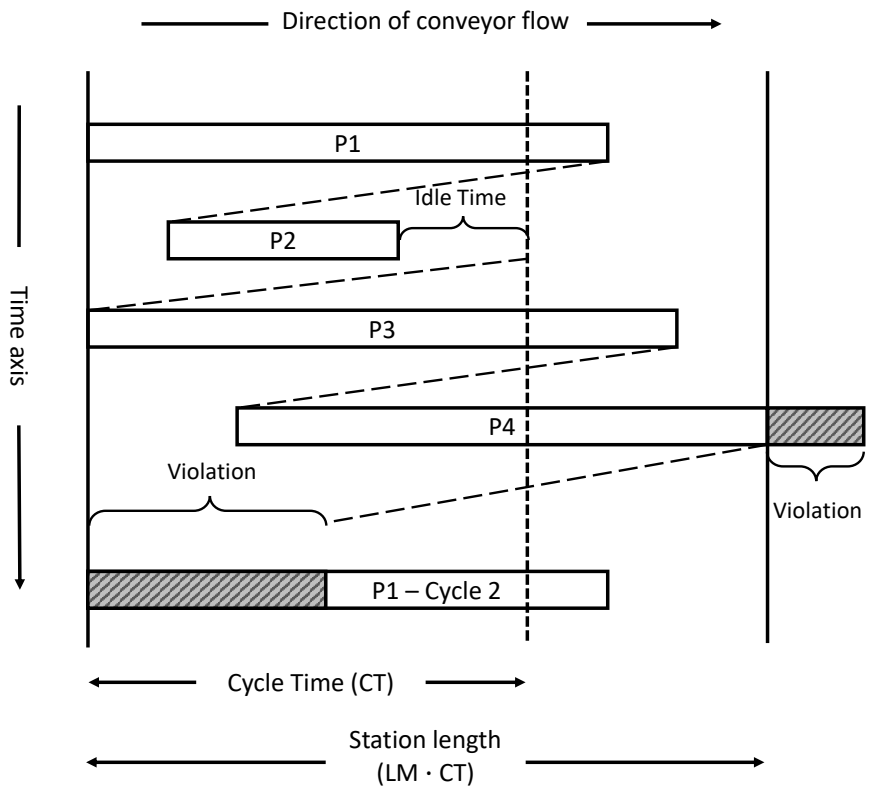

Figure 1: Example of the scheduling in a station of a paced assembly line.

present in the sequencing of the line during its lifetime. Following the same reasoning of recent works on balancing and sequencing of mixed-model assembly lines (Tiacci, 2015a; Öztürk et al., 2015; Lopes et al., 2018, 2019), the product mix is modeled as a minimal part set (MPS) in a cyclic sequencing. The MPS sequences can be considered as building blocks of a master sequencing. The cyclical aspect makes sure that the parts are repeatable and to some extend interchangeable. The objective of the balancing is to find the line configuration with the least expected utility work based on a given demand scenario set.

The described problem is formulated in an integrated balancing and sequencing model. The balancing decisions must be made a priori, while the sequencing decisions are based on the scenarios. The division of variables before and after the realization of a stochastic event is typical of stochastic two-stage problems (Birge & Louveaux, 2011).

The sets and the required data are defined in Table 2, while the variables are summarized in Table 3. A set T of tasks t is distributed into stations (set S) respecting precedence relations ($Prec$). The sequencing deals with a collection of product models M. Each model m exhibits task processing times $Dur_{t,m}$. The duration of a task of a given model can be set to zero if the product does not require such task. For the stochastic version of the problem, multiple demand scenarios $d \in D$ are considered with a probability of Pr_d. For each given demand scenario, the demand for each model ($Dem_{m,d}$) is given.

Based on each scenario d, a set of product pieces P_d is defined. The set P_d has an index d, because the number of pieces in a minimal part set (MPS) can differ based on the demand of each scenario. The cycle time (CT) is given and the station length is measured in relation to the equivalent distance of cycle time considering a conveyor speed of 1 and the constant length multiplier LM ($LM \cdot CT$).

The first-stage decisions are based on the balancing variables x. The second stage depends on binary sequencing variables y and continuous variables U, PT, and Pos to model the utility work, processing time, and final work position for every workpiece and station for all demand scenarios.

Table 2: Nomenclature of sets and data of the formulation.

Sets	**Meaning**
T	Set of tasks t
S	Set of stations s or k ordered along the belt
M	Set of product models m
D	Set of demand scenarios d
P_d	Set of pieces p of the demand scenario d
$Prec$	Set of precedence relations (t_1, t_2)
Data	**Meaning**
$Dur_{t,m}$	Duration time of task t of model m
$Dem_{m,d}$	Demand of model m in scenario d
Pr_d	Probability of scenario d
CT	Cycle time
LM	Length multiplier, used to define the maximal length of the stations

Table 3: Nomenclature of model variables.

Variable	**Domain**	**Type**	**Meaning**
x	(T, S)	Binary	1, if task t is assigned to station s, 0, otherwise
y	(P, M, D)	Binary	1, if the p^{th} piece is of model m for scenario d, 0, otherwise
U	(P, S, D)	$\mathbb{R}^+$	Utility work of piece p at station s and scenario d
PT	(P, S, D)	$\mathbb{R}^+$	Workload of piece p at station s and scenario d
Pos	(P, S, D)	$\mathbb{R}^+$	Final position of the worker for piece p at station s and scenario d

$$\textbf{Minimize} \sum_{d \in D} \sum_{s \in S} \sum_{p \in P_d} \frac{Pr_d \cdot U_{p,s,d}}{|P_d|} \tag{1}$$

$$\sum_{s \in S} x_{t,s} = 1 \qquad \forall\, t \in\ T \tag{2}$$

$$\sum_{k \in S : k \leq s} x_{t_1,k} \geq \sum_{k \in S : k \leq s} x_{t_2,k} \qquad \forall\, (t_1, t_2) \in Prec, s \in S \tag{3}$$

$$\sum_{t \in T} x_{t,s} \cdot Dur_{t,m} \leq LM \cdot CT \qquad \forall\, s \in S, m \in M \tag{4}$$

$$\sum_{m \in M} y_{p,m,d} = 1 \qquad \forall\, d \in D, p \in P_d \tag{5}$$

$$\sum_{p \in P_d} y_{p,m,d} = Dem_{m,d} \qquad \forall\ m \in M, d \in D \tag{6}$$

$$PT_{p,s,d} \geq \sum_{t \in T} x_{t,s} \cdot Dur_{t,m} - LM \cdot CT \cdot (1 - y_{p,m,d}) \qquad \forall\ d \in D, m \in M, s \in S, p \in P_d \tag{7}$$

$$\sum_{p \in P_d} PT_{p,s,d} = \sum_{t \in T} \sum_{m \in M} x_{t,s} \cdot Dur_{t,m} \cdot Dem_{m,d} \qquad \forall\ d \in D, s \in S \tag{8}$$

$$Pos_{p,s,d} \geq Pos_{p-1,s,d} - CT + PT_{p,s,d} - U_{p,s,d} \qquad \forall\ d \in D, s \in S, p \in P_d \mid p > 1 \tag{9}$$

$$Pos_{1,s,d} \geq Pos_{|P_d|,s,d} - CT + PT_{1,s,d} - U_{1,s,d} \qquad \forall\ d \in D, s \in S \tag{10}$$

$$Pos_{p,s,d} \geq CT \qquad \forall\ d \in D, s \in S, p \in P_d \tag{11}$$

$$Pos_{p,s,d} \leq LM \cdot CT \qquad \forall\ d \in D, s \in S, p \in P_d \tag{12}$$

$$x_{t,s}, y_{p,m,d} \in \{0, 1\} \tag{13}$$

$$U_{p,s,d}, PT_{p,s,d}, Pos_{p,s,d} \in \mathbb{R}^+ \tag{14}$$

The monolithic model is presented by the expressions 1-14. Its objective function is the minimization of the expected utility work per unit produced. Expression 1 is modeled as a weighted average of the utility work on all scenarios divided by the number of products in each scenario. Expressions 2 and 3 are the occurrence and precedence restrictions (Ritt & Costa, 2018). The balancing part of the model is completed with ineq. 4. So that the products can be processed within the stations, the workload of any station cannot be longer than the station length ($LM \cdot CT$).

The sequencing part of the model assigns product models to workpieces. The assignment restrictions are modeled by eqs. 5 and 6. Every workpiece must be one of the product models (eq. 5), while the number of pieces of a given model must be equal to the demand for a given scenario (eq. 6). The processing time variables (PT) integrate the balancing and sequencing. Ineq. 7 is a Big-M constraint assigning the processing time of workpieces to the corresponding duration of their model tasks. The sum of the processing time is also enforced by eq. 8, which tightens the linear relaxation of the model.

Expressions 9-12 model the scheduling of paced assembly lines. Variable Pos controls the final position of a worker after working on a piece. This position is determined by ineq. 9 based on the position of the last piece ($Pos_{p-1,s,d}$), the time distance between the two pieces (CT), the processing time of the piece ($PT_{p,s,d}$), and the employed utility work ($U_{p,s,d}$). Ineq. 10 represents the cyclical sequencing link of the first and the last previous of two different replications of the MPS. Further bounds complete the formulation of the problem to assure no task is performed outside of the station borders. Ineq. 11 assures that the initial position of a piece ($Pos_{p,s,d} - CT$) cannot be negative, as the worker can return the equivalent of CT at the line. For the maximal bound, the final work position must be within the station limit (ineq. 12). Finally, expressions 13 and 14 present the binary restrictions on variables x and y and the non-negativity of the continuous variables.

Notice that the balancing restrictions are independent of the demand scenario (no index d), while the sequencing part of the model has variables and expressions to every scenario. This structure is commonly found in stochastic problems and it is explored in the decomposition approach presented in Section 4.

4. Combinatorial Benders' decomposition

4.1. Benders' decomposition

The Benders decomposition (Benders, 1962) or L-shaped method (Van Slyke & Wets, 1969) is a procedure in which problem variables are split into exclusive subsets so that the solution of its parts does not have to deal simultaneously with all problem constraints. This method decomposes the two (or more) stages into different formulations, called master problem (MP) and subproblem (SP) (Birge & Louveaux, 2011; Rahmaniani et al., 2017). For stochastic problems, each realization of the random variables is modeled as an SP. In the MP, the information of random realizations is approximated with a variable θ. The decomposition consists in the iterative solution of MP and SPs, in which the information of the SPs is used to improve the approximation provided by θ with the use of cuts.

The standard L-Shape Method uses the dual information of the SPs to generate the cuts (Laporte & Louveaux, 1993; Birge & Louveaux, 2011). When the SPs contain only continuous variables, the dual information is enough to approximate the expected objective function with θ. If the subproblems contain integer variables, however, the dual information of the solution is not as meaningful as in continuous problems and other cut generation schemes must be used.

One of the approaches that use cuts based on the integrality conditions of binary variables is given by Laporte & Louveaux (1993). In a formulation with binary variables, for a given solution in an iteration k, S_k is defined as the set containing the variables with value of 1. In

$$\theta \geq (\theta^k - L)\left(\sum_{i \in S_k} x_i - \sum_{i \notin S_k} x_i - |S_k|\right) + \theta^k \tag{15}$$

θ is set to be equal or larger than the objective function of the subproblem realization (θ^k) for a given iteration. Cut 15 is a combinatorial cut and assures that θ assumes the realized value when $\sum_{i \in S_k} x_i = |S_k|$ or is relaxed to a restriction $\theta \geq L$ when one of the variables differs from the solution. The parameter L is a lower bound for the expected value of the subproblem realizations. Laporte & Louveaux (1993) also provides a strengthening formulation for the cut. However, the combinatorial cuts enforce lower bounds only locally, that is, for a specific combination of binary variables.

Another set of combinatorial Bender cuts can be used for binary variables if the subproblem works as a feasibility check. Codato & Fischetti (2006) present a decomposition scheme for mixed-integer linear problems with objective function based either only on MP's variables or only on SPs' variables. When only the MP variables have a direct influence on the objective function, the subproblems are used to check whether the solutions of the MP are feasible. In the case of an infeasibility, a cut in the form

$$\sum_{i \in C} x_i \leq |C| - 1 \tag{16}$$

can be applied. For this cut, $C \subseteq X$ is a minimal infeasible subsystem (MIS) that causes the infeasibility (Codato & Fischetti, 2006), while X is the set of variables. Cut 16 states that, at least one of the variables

must be different than the ones in set C, otherwise the solution is infeasible. For the case in which the objective function is only defined based on the SP, the SP can be reformulated with an upper (or lower) bound for the objective function. That is, a solution that improves the objective function on at least one unit is searched. If no solution exists, the subproblem is said to be infeasible. With the new bound, the SP also works as a feasibility check (Codato & Fischetti, 2006). This way, a combinatorial cut is generated in the MP cutting off combinations that cannot improve the solution. The authors state that the cut may not work efficiently for all applications, as the strength of the cuts depends on the combinatorial structure of the problem.

In the balancing context, the Benders Decomposition is found in the papers of Bentaha et al. (2014), Akpinar et al. (2017), Michels et al. (2019), and Michels et al. (2020). Bentaha et al. (2014) consider the balancing of disassembly lines, in which the task times are stochastic. The realization of the dismounting operations is then modeled in a continuous subproblem. In this problem, the complexity lays not in solving one realization of the second stage, but in the exponential number of scenarios of processing times with respect to the number of tasks. The authors mitigate the problem by using a sample average approximation to limit the number of computed scenarios. Akpinar et al. (2017) uses combinatorial cuts based on ineq. 16 for the balancing of an assembly line with setup times. The problem is decomposed into an MP dealing with the balancing decision and SPs deciding the sequencing of tasks inside the stations. The SPs work as a feasibility check: an optimal solution of the MP may be either feasible and optimal for the whole problem or infeasible due to the setup times. The procedure is able to solve instances to optimality with up to 58 tasks. Michels et al. (2019) use a similar approach for the balancing of assembly lines with multiple workers. As the operations of a station may be divided between workers, a scheduling of the tasks is solved as the SP of the Benders decomposition. The combinatorial cuts added in the MP either cut out infeasible assignments or require an extra worker to the station. The approach can solve instances of the Multi-manned Assembly Line Balancing Problem (MALBP) with up to 148 tasks. A version of the algorithm for the type-2 problem (minimization of the cycle time) is proposed in Michels et al. (2020).

In this section, a Benders Decomposition for the proposed problem (Section 3) is described. The stochastic nature of the demand presents a clear block structure. Hence, the sequencing of each demand scenario can be solved independently from the others. A decomposition based on the stations, as presented in Akpinar et al. (2017), is not directly possible. In an assembly line, the flow of the products is performed by a conveyor belt. It follows that the sequence of products is the same for all stations. The combinatorial cuts are then based on the set of task assignment variables for all stations. These cuts work only locally, so that valid inequalities and partially relaxed cuts are also introduced in the proposed solution method.

4.2. Decomposition structure

The monolithic model (Expressions 1-14) is decomposed into one Master Problem (MP), containing the balancing decisions, and Subproblems (SP). The MP inherits the binary x variables and the Expressions 2-

4. Additionally, a variable $\theta_{d,s}$ is defined as an approximation for the utility work of the demand scenario d at station s. The objective function of the MP is then the weighted average sum of approximation variables

$$\textbf{Minimize} \sum_{d \in D} \sum_{s \in S} \frac{Pr_d \cdot \theta_{d,s}}{|P_d|} \tag{17}$$

to minimize.

The MP is complemented with a set of cuts K. Inequality

$$\sum_{s \in C_k} \theta_{d,s} \geq \theta_d^k \cdot \left(\sum_{(t,s) \in S_k} x_{t,s} - |S_k| + 1 \right) \qquad \forall\ k \in K, d \in D \tag{18}$$

is an adaptation of the lower-bounding function of Laporte & Louveaux (1993). On the left-hand-side, instead of the single approximation variable as in ineq. 15, a sum of the related $\theta_{d,s}$ values is used. The definition of the index s in the θ variables is necessary for the generation of partial combinatorial cuts, which are further introduced in Section 4.4. The sum operator is defined on base of the set C_k, containing a subset of stations. On the right-hand-side, a few simplifications are made in ineq. 15. Firstly, the lower bound L on utility work is assumed to be 0, since obtaining the real bound may require solving several sequencing problems. The set S_k contains all pairs (t, s) for which $x_{t,s} = 1$ and $s \in C_k$. As the $x_{t,s}$ variables represent assignment of tasks to stations, an extra task can only increase the utility work necessary in a station. Therefore, it is not needed to subtract $\sum_{(t,s) \notin S_k} x_{t,s}$ in ineq. 18, as it is in ineq. 15. As SPs can be solved independently, an index d refers to each demand scenario, with their objective function value θ_d^k.

The SPs are related to the sequencing of models in the assembly line. The Expressions 5 - 14 defines the sequencing problem. For the SP, however, the index d is left out, since one subproblem is defined for every given demand scenario d. The objective function

$$\textbf{Minimize} \sum_{p \in P_d} \sum_{s \in S} U_{p,s} \tag{19}$$

is the minimization of the necessary utility work for the sequencing of models.

4.3. Tightening the formulation: preprocessing and valid inequalities

In the integer version of the Benders decomposition, a tighter formulation is important to reduce the integrality gap and consequently the number of explored nodes (Fischetti et al., 2017). In this section, the preprocessing of the balancing and sequencing problems is explained along with the development of valid inequalities.

4.3.1. Balancing preprocessing

A preprocessing or variable elimination for the balancing problem is introduced by Patterson & Albracht (1975). Using the precedence relations and the processing times, bounds on the position of a task in the line can be defined. The Earliest Station (E_t) and Latest Station (L_t) of a task t for a single model

can be specified by as how many stations are necessary to pack all the direct and indirect predecessors of a given task (and successors, for the Latest Station).

For the preprocessing of the proposed problem, the feasible interval of assignments is adapted considering the multiple models and demand scenarios. First, E_t^{avg} and L_t^{avg} are defined as the earliest and latest station based on the tasks' average durations. The average task duration is calculated as the weighted average of processing time weighted to the demand and probability of each scenario. In this calculation, it is assumed that the line capacity (based on the cycle time) is large enough to produce the average processing times without utility work, otherwise, the line would be undersized.

Other valid bounds can be found by computing the models individually. From ineq. 4, it was assumed that no model requires more processing time in a station than its length ($LM \cdot CT$). For the case of tasks longer than the cycle time (or possibly the station length), approaches such as multiple workers or parallel stations are used, which is not the case in the proposed problem. Individually, a model-specific Earliest Station

$$E_t^m = \left\lceil \frac{Dur_{t,m} + \sum_{(i,t)\in Prec^+} Dur_{i,m}}{CT \cdot LM} \right\rceil \qquad \forall\, t \in T, m \in M$$

and Latest Station

$$L_t^m = |S| + 1 - \left\lceil \frac{Dur_{t,m} + \sum_{(t,i)\in Prec^+} Dur_{i,m}}{CT \cdot LM} \right\rceil \qquad \forall\, t \in T, m \in M$$

can be calculated, considering the station length as the loading limit of each station and $Prec^+$ is the set of all transitive precedence relations.

The set of feasible stations for each task can be computed as the intersection of stations between the average and the model-specific earliest and latest stations. Using the preprocessing, only a subset of balancing variables have to be considered. Note that for each model the preprocessing is not very powerful: the limit of the cycle time for each station is the length of a station ($LM \cdot CT$) compared to the cycle time CT. The intersection of all models, however, can result in a larger domain reduction.

4.3.2. Valid inequalities based on the linear relaxation

Another approach to improve the MP are valid inequalities (Rahmaniani et al., 2017). In the decomposition, the MP loses information that is based on the second-stage. In some cases, valid inequalities can be used to further restrict the MP and improve the approximation of the variable θ. Rahmaniani et al. (2017) report that one common strategy is to solve the linear relaxation of the SP and add cuts based on the linear relaxation solution (LP-based cuts). For the presented problem, the regular Benders' cut based on the relaxation of the sequencing problem is

$$\theta_{d,s} \geq \sum_{t\in T} \sum_{m\in M} Dur_{t,m} \cdot Dem_m \cdot x_{t,s} - |P| \cdot CT \quad \forall\, d \in D, s \in S. \tag{20}$$

The development of the expression based on the shadow prices of the SP is detailed in the online appendix. The relaxation of a sequencing problem with binary assignment variables has only utility work if the total

processing time assigned to a station ($\sum_{t\in T}\sum_{m\in M} Dur_{t,m}\cdot Dem_m\cdot x_{t,s}$) is larger than the available time ($|P|\cdot CT$) for the processing of $|P|$ pieces on every station s.

4.3.3. Unavoidable idle time

Depending on the line length, it is possible to infer idle times based only on the balancing variables of the MP. As every piece flows CT units of time away from each other, short processing times for a specific model may result in idle time. The continuous and non-negative variable $I_{m,s}$ is used to capture this unavoidable idle time, which is modeled by

$$I_{m,s} \geq (2 - LM)\cdot CT - \sum_{t\in T} Dur_{t,m}\cdot x_{t,s} \quad \forall\ m\in M, s\in S. \tag{21}$$

Independently of the sequence, idle time can be determined if the processing time is short enough so that after two cycles the worker necessarily has to wait for the next piece. Ineq. 21 is valid for the extreme case in which the previous product is at the last possible position of the line, as illustrated in Fig. 2. The worker returns CT units after processing $P1$ and goes to $P2$. If the processing time of $P2$ is not enough to reach the equivalent length of CT units, the worker would inevitably have to wait for the next piece, that is, idle time occurs. Cut 21 is only meaningful if the line length ($LM\cdot CT$) is shorter than the double length of the cycle time.

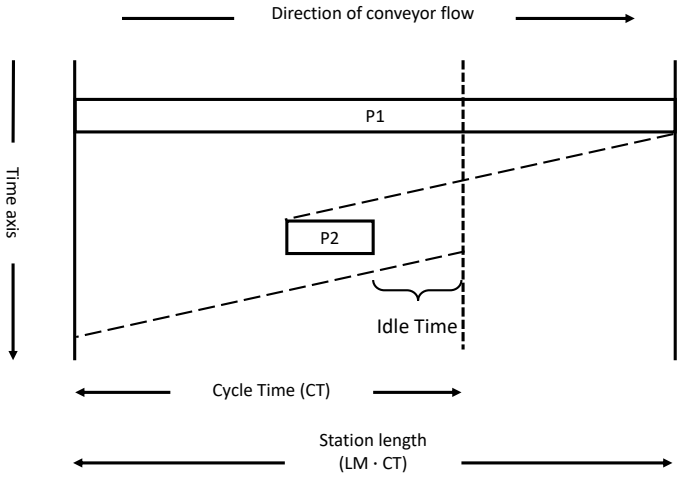

Figure 2: Example of assignment that invariably causes idle time.

The unavoidable idle times due to balancing ($I_{m,s}$) can be used to strengthen the bound of ineq. 20. Expression

$$\theta_{d,s} \geq \sum_{t\in T}\sum_{m\in M} Dur_{t,m}\cdot Dem_m\cdot x_{t,s} + \sum_{m\in M} Dem_m\cdot I_{m,s} - |P|\cdot CT \quad \forall\ d\in D, s\in S \tag{22}$$

is an improved version of the LP-based cut. The cut forces $\theta_{d,s}$ to assume utility work if the sum of the processing times plus the unavoidable idle times are larger than the available time in a station.

4.4. Partial combinatorial cuts

Even with the preprocessing and the valid inequalities, the MP still presents relatively large integer gaps. The standard procedure to close the gap is based on combinatorial cuts, as in Laporte & Louveaux (1993); Codato & Fischetti (2006). These cuts, such as in ineq. 18, applied to all stations are, however, local and their efficiency is highly based on the problem structure (Codato & Fischetti, 2006).

In this work, an alternative cutting generation strategy is proposed. Along with the local cut for each scenario (as in ineq. 15) that cuts mainly the explored SP, the SPs are further decomposed. In the decomposition, sequencing problems with only a subset of the stations are solved. Although the sequence of pieces must be the same in all stations, relaxing this restriction results in lower bounds for the sequencing problem. These lower bounds provide valuable information if the processing times in one or a subset of stations causes utility work independent of the others. A low-density cut (based on a few variables) can then be added to the MP using the ineq. 18 containing only the subset of stations that has relevant utility work.

The function of the cut is shown with the help of a numerical example given in Table 4. An assembly line with three stations and four models (M1-M4) with only one scenario of demand of one unit each is considered. The cycle time is five time units, while the length of each station is equivalent to seven time units. The processing time of eight tasks and three balancing solutions are represented in Table 4.

Table 4: Task durations and solutions for an example instance. The columns 'Solution' contains the station to which each task is assigned.

	Processing Time				Solution		
Task	M1	M2	M3	M4	S1	S2	S3
1	4	2	2	4	1	1	1
2	2	1	1	3	1	1	2
3	3	2	1	1	2	2	1
4	4	4	3	2	2	2	2
5	1	1	2	3	3	3	3
6	2	3	4	4	3	4	3
7	5	4	2	2	4	3	4
8	2	3	1	1	4	4	4

The first solution (S1) is feasible in the MP and all θ values are zero, meaning that no utility work is expected. The solution of the sequencing subproblem, however, has an optimal cyclic sequence $1-3-2-4$ with one unit of utility work. The combinatorial cut based on the subproblem is

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 \geq 1 \cdot (x_{1,1} + x_{2,1} + x_{3,2} + x_{4,2} + x_{5,3} + x_{6,3} + x_{7,4} + x_{8,4} - 8 + 1). \tag{23}$$

A possible solution for the MP after the addition of the cut 23 is represented by the second solution

in Table 4 (S2). This balancing solution has θ as zero and is not affected by the cut 23. Its optimal cyclic sequencing solution, however, is also $1-3-2-4$ with one unit of utility work. The local cut 23 is not able to cut off neighboring solutions that may be as bad as the one solved previously.

An alternative to generating high-density cuts (cuts based several variables) is to solve the sequencing problem for a subset of stations. In solution one, every station, individually, has a feasible sequence with no utility work ($1-2-4-3$ for the first station and $1-3-2-4$ for the others). The sequencing problem considering only the first and second stations has an optimal sequence solution ($1-3-2-4$) with utility work of one unit. This information is relevant because a more specific cause of the utility work is now known: the interaction between the sequencing of the first and second stations. A low-density cut considering only the two stations is represented in

$$\theta_1 + \theta_2 \geq 1 \cdot (x_{1,1} + x_{2,1} + x_{3,2} + x_{4,2} - 4 + 1). \tag{24}$$

The second solution in Table 4 would also be affected by the low-density cut 24. A new iteration would force the change in one of the assignments of the first or second station. The third solution of Table 4 is an optimal solution for the problem, as the sequence $1-3-2-4$ does not present utility work for the processing times based on this balancing solution.

The partial combinatorial cuts are useful if causes of inefficiencies are observed in subsets of the problem. The low-density cuts can be less powerful than the normal combinatorial cuts for the solution used to generate them, as they are only a lower bound on the subproblem solution. However, the information on part of the problem provides valuable estimates of utility work for the MP and can affect other assignments, accelerating the solution. A related approach is reported by Fischetti et al. (2016). They also propose cuts based on the relaxation of the subproblem by dropping capacity constraints. In their approach, called blurring, the relaxed subproblems also produce lower bounds and are obtained faster than the regular cuts.

The partial decomposition may be applied to other problems when the partition of the problem variables represents a lower bound of the problem (for minimization problems). In the sequencing subproblem, decomposing the sequencing based on stations provides a lower bound and can be used to generate cuts. Despite sequencing problems based on a subset of the stations having the same complexity as the complete problem, solving them for a small subset of stations is much faster than the whole subproblem.

4.5. Proposed algorithm

The Benders' decomposition works iteratively between the master and subproblems. In the original formulation by Benders (1962), both the MP and SPs are solved to optimality in each iteration. Reports on the survey by Rahmaniani et al. (2017) state that for several applications the solution of the MP can take more than 80% of the solving time. Universal solvers provide a possibility that Fischetti et al. (2017) describe as a "modern implementation" of the Benders' Decomposition. The MP is solved just once. Every time an incumbent solution is found, the subroutines that generate the Benders cuts based

on the SP are called. This approach generates the Benders Cuts "on the fly" and is used to speed up the algorithm (Codato & Fischetti, 2006; Costa et al., 2012; Akpinar et al., 2017; Fischetti et al., 2017; Michels et al., 2019).

The structure of the algorithm is described in Fig. 3. The MP is solved in a single tree and for every incumbent solution found, a callback is used to check the solution and add cuts. The solution is checked at two levels. In the first level, sequencing problems with subsets of the stations are solved. This way, assignment inefficiencies that result in utility work can be identified. The combinations of stations that generate utility work are used in combinatorial cuts for the MP. If the observed utility work is larger than the best solution found by then, the solution is cut-off and the algorithm returns to the MP. If the solution cannot be cut-off, the complete SPs are solved. Based on the solutions, cuts are added in the MP and the incumbent objective function value can be updated. If the utility work of the solution is equal to the objective value of the MP, the algorithm converges and the optimal solution is found.

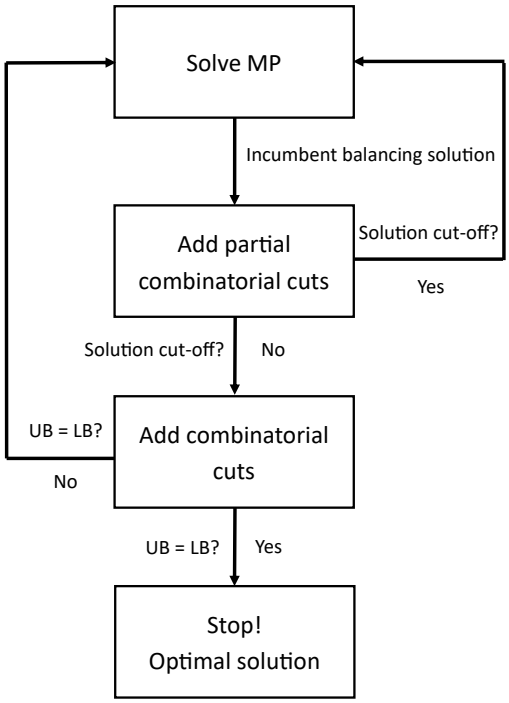

Figure 3: Structure of the Benders Decomposition Algorithm.

The difference between the proposed algorithm and standard Benders' decomposition structure lies in the partial combinatorial cuts. Without these cuts, the algorithm iterates between MP and the SPs for each demand scenario d until convergence is reached. The extra block "Add partial combinatorial cuts" in Fig. 3 is intended to speed up the convergence. There is a trade-off between the number of combined stations and the time invested generating the partial cuts. Therefore the number of selected stations and how they are combined is important for the solution efficiency. The details of the implementation are described in Section 5. A proof of the algorithm convergence is given in the online appendix.

5. Computational study

In this section, the proposed Benders' decomposition algorithm is compared with the monolithic MILP formulation of the problem. The dataset generation is described in Section 5.1, where three datasets are proposed. The algorithm implementation details are described in Section 5.2. An analysis of the algorithm features is given in Section 5.3, in which a parametric dataset is solved for several algorithm configurations. In Section 5.4, the stochastic solution is compared to the deterministic model based on the average demand for a dataset with large instances under real-world demand inspired scenarios. A subset of instances are solved for longer MPS and more demand scenarios in Section 5.5 to observe the effect of these parameters. Finally, Section 5.6 contains the discussion of the results and the managerial insights.

5.1. Instance generation

There is no consensus on standard datasets for the Mixed Model Assembly Line Problems (Lopes et al., 2020). Most of the authors create their instances or adapt the structured datasets for the extra characteristics. In this paper, the generation of processing time data is based on Lopes et al. (2018). That is, several single-model instances are used to generate a mixed-model instance.

The created datasets are built based on ordering strength (OS), average processing load (PL), and line length multiplier (LM) parameters. OS is a measure of how restricted the precedence diagram is. The value varies between 0 (no restrictions) to 1 (a single possible sequence) (Otto et al., 2013). The OS values of 0.2, 0.6, and 0.9 are taken from the dataset of Otto et al. (2013). The PL parameter takes part in the calculation of the cycle time

$$CT = \left\lceil \frac{\sum_{t \in T} \sum_{m \in M} \sum_{d \in D} Dur_{t,m} \cdot Dem_{m,d} \cdot Pr_d}{PL \cdot NS} \right\rceil.$$

The numerator is the expected processing time of all models in all demand scenarios, while the denominator is the number of stations and a multiplier. A multiplier PL of 90%, for instance, represents that the expected average occupancy of the line divided between all stations and the utility workers is 90%. Finally, the length multiplier (LM) defines the station length in terms of the cycle time. A value of 150% and a cycle time of 1,000 time units, for instance, result in a station length equivalent to 1,500 time units (considering a conveyor speed of 1 unit of distance per unit of time).

5.1.1. Dataset 1: a parametric dataset

This first dataset is constructed based on instances of the same size to test the influence of OS, PL, and LM parameters and the respective algorithm performance for each parameter constellation. The parameters used for the generation of the instances are described in Table 5. The processing times and precedence relations are based on the medium instances (50 tasks) of the standard dataset from Otto et al. (2013). The number of models (NM) in each instance is set to 10 so that every instance of the dataset is generated based on 10 single-model instances. The dataset and the list of the base instances

from Otto et al. (2013) are described in the online supplementary material. All instances are generated with 10 stations and 5 demand scenarios. The demand scenarios are considered of given probability (0.2) and their model demand is given in Table 6. The demand scenarios are similar to the ones proposed by Chica et al. (2013), considering a uniform demand scenario and scenarios in which families of models have increased demand.

Table 5: Parameters used in the generation of the dataset.

Parameter	Values
Number of tasks (NT)	50
Number of models (NM)	10
Number of stations (NS)	10
Number of demand scenarios (ND)	5
Ordering strength (OS)	0.2 / 0.6 / 0.9
Average processing load (PL)	90% / 95%
Length multiplier (LM)	120% / 150% / 200%

Table 6: Model demand for each of the scenarios in every instance of the dataset.

	Model No.										
Scenario	1	2	3	4	5	6	7	8	9	10	Prob.
1	1	1	1	1	1	1	1	1	1	1	0.2
2	3	3	2	2	0	0	0	0	0	0	0.2
3	0	0	0	0	4	3	3	0	0	0	0.2
4	2	2	2	1	0	0	0	1	1	1	0.2
5	0	0	0	0	2	2	2	2	1	1	0.2

Five instances are created for each combination of parameter values (OS, PL, and LM). As the combination $OS = 0.9$ (low assignment flexibility) with $LM = 120\%$ (low sequencing flexibility) produced infeasible instances, they are not considered in the dataset. The resulting dataset has 80 instances and is available at the supporting information.

5.1.2. Dataset 2: inspired on approximated reported monthly sales

A second dataset containing larger problems in terms of number of tasks and stations is also proposed. This second dataset is based on the instances from Scholl (1993), containing 25 precedence graphs ranging from 8 to 297 tasks. One instance is generated for each of the 25 precedence graphs.

For the generation of models and demand scenarios, data on the number of licensed vehicles of a manufacturer in Brazil is used. FENAFABRE (2019) (Brazilian National Federation of Automotive Vehicle Distributors) publishes monthly statistics on the number of licensed automobiles per manufacturer and per model. For the generation of the dataset, the monthly data of 2019 is used. The reports and a spreadsheet with the data can be found in the online supplementary materials. Although the number of licensed vehicles do not match perfectly with the production, this data is used as a proxy for the demand

levels. In practice, these values do not consider lost demand and production exported to other countries (about 14.5% in 2019 (ANFAVEA, 2019)).

The chosen manufacturer produces five commercial vehicles in a single facility, whose reported data is used to approximate the demand scenarios. Table 7 contains the number of licensed vehicles for each month and the production mix used for the instances. For each month, an MPS is calculated so that the sum of the square error from the real data is minimized for sequences of up to 10 products. The generation of the most representative MPS is given by the formulation in Expressions 25-27. Dem_m^{real} is the real demand of model m (approximated with sale numbers), while Dem_m^{MPS} is the best approximation of the demand of model m for an MPS of most 10 products. The real demand vectors and the 12 obtained demand scenarios are given in the right side of Table 7.

$$\textbf{Minimize} \sum_{m \in M} \left(\frac{Dem_m^{real}}{\sum_{k \in M} Dem_k^{real}} - \frac{Dem_m^{MPS}}{\sum_{k \in M} Dem_k^{MPS}} \right)^2 \tag{25}$$

$$\sum_{m \in M} Dem_m^{MPS} \leq 10 \tag{26}$$

$$Dem_m^{MPS} \in \mathbb{N} \quad \forall\, m \in M \tag{27}$$

Table 7: Monthly demand of the chosen manufacturer in 2019 and the approximated MPS used.

Month /	No. of licensed vehicles					Approximated MPS				
Model No.	1	2	3	4	5	1	2	3	4	5
Jan	5336	3650	2557	1911	1078	4	2	2	1	1
Feb	5473	3604	1746	1686	1786	4	3	1	1	1
Mar	5853	4334	1872	1892	3661	3	2	1	1	2
Apr	7319	3927	1871	2015	2275	4	2	1	1	1
May	8661	3834	1841	2490	2273	5	2	1	1	1
Jun	7882	3007	3898	3019	1625	4	1	2	2	1
Jul	8070	2954	2680	1777	1272	5	2	1	1	1
Aug	7455	5347	2073	1253	2280	4	3	1	1	1
Sep	8826	5355	2729	2055	2014	4	3	1	1	1
Oct	6066	4288	2799	2075	2394	3	2	2	1	1
Nov	6009	4388	2837	2749	3227	3	2	1	1	2
Dec	8174	5608	1760	3171	3120	4	3	1	1	1
Total	85124	50296	28663	26093	27005					

The processing times of each of the 25 instances are used as base values to the generation of the five models in production. Table 8 contains the parameters used for the random generation of models. Each task has a probability of not being required (column 'Not required'), of being equal to the base value ('Equal'), or varying within a uniform distribution. Four intervals are used: reduction of 25% to 50%, small variations within -25% and +25%, and increases of [25%, 50%] and [50%, 100%]. The probabilities are chosen to correspond to the complexity of the real models. The more complex models 3 and 4, for instance, have higher probabilities of longer processing times.

Table 8: Parameters used for the random generation of the models processing times.

Model No.	Equal	[-25%, -50%]	[-25%, +25%]	[+25%, +50%]	[+50% , 100%]	Not required
1	0.7	0	0	0	0	0.3
2	0.3	0.1	0.2	0.1	0.1	0.2
3	0.15	0	0.2	0.2	0.3	0.15
4	0.15	0	0.3	0.2	0.2	0.15
5	0.2	0.1	0.4	0.1	0	0.2

The 25 instances are generated with PL (average processing load) of 0.95 and an LM (station length multiplier) of 1.5. The number of stations is defined as an average of all instances of Scholl (1993) dataset. If any instance was proven infeasible, new random values for the time distributions were drawn. The instances with their number of tasks, number of stations, and the cycle time used are summarized in Table 9.

Table 9: Number of tasks, stations, and the cycle time for each instance of the second dataset.

Instance	No. Tasks	No. Stations	Cycle time
Arcus1	83	12	5568
Arcus2	111	15	9157
Barthold	148	9	605
Barthol2	148	39	96
Bowman	8	5	15
Buxey	29	10	30
Gunther	35	10	44
Hahn	53	6	2195
Heskiaoff	28	5	199
Jackson	11	5	9
Jaeschke	9	6	8
Kilbridge	45	7	76
Lutz1	32	10	1407
Lutz2	89	18	25
Lutz3	89	13	114
Mansoor	11	3	56
Mertens	7	4	8
Mitchell	21	5	21
Mukherje	94	14	291
Roszieg	25	7	18
Sawyer	30	10	32
Scholl	297	38	1610
Tonge	70	14	249
Warnecke	58	16	91
Wee-Mag	75	16	92

5.1.3. Dataset 3: testing the number of demand scenarios and MPS' length

A third dataset containing 10 instances is used to explore the effect of the number of demand scenarios and the length of the MPS. For this dataset, the precedence graph of 'Hahn' is used (Scholl, 1993). The

generation of the models is identical to the second dataset, but the number of demand scenarios is increased. The instances are created using an MPS of 10 and 15 products for 10, 20, 30, 40, and 50 demand scenarios resulting in 10 instances. The demand scenarios are generated randomly based on a uniform distribution of the average yearly vehicle licensing data. That is, every model is selected randomly with a probability of 39.2% for model 1, 23.2% for model 2, 13.2% for model 3, 12.0% for model 4, and 12.4% for model 5, according to the total production in 2019 given in Table 7. Every 10 (or 15) models are combined in an MPS, which represents one possible demand scenario in the instance. Values of 95% and 150% for PL and LM are used. All instances are available in the online supplementary material.

5.2. Implementation details

The decomposition is implemented in Visual Basic 15.0 and the solver Gurobi 8.1 is used to solve the master problem (MP), subproblem (SP), and the monolithic model. All tests are run on an Intel i7 8700K processor with 6 cores at 4.0 GHz using 32 GB of RAM.

In the implementation, the MP is solved only once using Gurobi. For each incumbent solution (feasible task assignment with estimative values for the utility work) found SPs are solved to determine the real value of the utility work. The objective value of the SPs is then relayed to the MP via cuts (ineq. 18) implemented as 'lazy cuts' (Fischetti et al., 2017).

For the complete version of the algorithm, the improvements of Section 4.3 (preprocessing and valid inequalities) are included in the formulation. In order to apply the partial combinatorial cuts of Section 4.4, subsets of stations must be selected. Every time that the optimal sequencing of a subset of stations requires more utility work than estimated in the MP for these stations, a combinatorial cut can be defined.

For the results of Section 5.3, partial SPs with groups of up to three stations are selected. As a level 1 ($L1$) SP, the sequencing of all individual stations is solved. For multiple stations, the selection is made in order to provide a cover of all stations instead of considering all the possible combinations. For simplicity, adjacent stations are chosen to define the SPs. Therefore Level 2 ($L2$) SPs are solved for stations 1 and 2; 3 and 4; etc. while Level 3 ($L3$) SPs correspond to stations 1, 2, and 3; 4, 5, and 6; etc. Although the complexity of solving the partial problems is the same as for the complete SP, the instances with up to 3 stations ($L1$ to $L3$) are solved quickly with modern solvers and can be dealt with in parallel.

5.3. Model comparison

To the best of the author's knowledge, there is no other method published for the proposed problem. Therefore, the monolithic model is used as the benchmark for the decomposition. In order to observe the effect of proposed cuts and preprocessing, the dataset is solved for several versions of the algorithm. The tested elements are the preprocessing of Section 4.3.1 (Pre), the LP-based cut of Section 4.3.2 (LP), the unavoidable idle time cut from Section 4.3.3 (Id), and the partial combinatorial cuts from Section 4.4 (Part).

Table 10 contains the results for the monolithic model and the proposed algorithm with all its features along with a combination of versions without some of the features. The column $Method$ describes which features are enabled for each run. Every line of the table contains the average results of 80 instances solved with a time limit of 3,600 seconds (as some methods did not find any feasible solution for 3 instances, the average upper bound information is based on 77 instances only). The average solution time (bounded by 3,600 seconds) is displayed divided in the time used to solve the MP and the time required for the SPs. Furthermore, the average number of combinatorial cuts added during the procedure are displayed in the $\#Cuts$ column. The $Full$ column refers to the standard combinatorial cut (eq. 18), while the remaining columns show the number of valid cuts generated solving each station individually ($L1$) and in combination of 2 or 3 stations ($L2\&3$). The columns $\#Nodes$ describe the number of solutions that are found by the MP. The nodes are divided in cut-off nodes and incumbent solutions (Inc). The nodes that are discarded based on the partial combinatorial cuts are further classified based on the class of the cut ($L1$, $L2\&3$, and $Full$).

Table 10: Summary of the results of the algorithm with the combination of its features (LP stands for the LP-based cut; Id the idle time cut; Pre the preprocessing; and $Part$ the partial combinatorial cuts). The average upper and lower bounds (UB, LB) are displayed along with the number of proven optimal instances (Opt). The average solution time is divided in master problem and subproblem (Sub). The $\#Nodes$ columns bring the number of solutions found in the master problem and whether they are cut off. Columns $\#Cuts$ display the average number of generated cuts. The data for the upper bound is the average of 77 instances, since some methods could not find feasible solutions for 3 instances.

	Method							Sol. time		# Nodes				# Cuts		
	LP	Id	Pre	Part	UB	LB	Opt	Master	Sub	L1	L2&3	Full	Inc	L1	L2&3	Full
1					417.21	0	1	859.1	2,733.5	-	-	16,211	16.0	-	-	44,506
2				x	54.63	14.33	42	1,712.5	362.2	1414	0.9	4.1	22.5	23,087	20.9	20.5
3			x		513.91	0	0	907.1	2,693.8	-	-	15,966	15.3	-	-	46,523
4			x	x	54.43	14.33	47	1,690.6	356.1	1466	1.2	4.8	22.4	23,391	20.6	21.4
5	x				19.96	16.27	64	667.0	221.1	-	-	249.0	15.2	-	-	360.7
6	x			x	20.65	16.06	65	796.1	59.1	4	1.2	8.4	14.7	50.2	22.2	27.1
7	x		x		20.08	15.69	61	735.1	225.4	-	-	188.3	15.4	-	-	292.8
8	x		x	x	20.27	15.54	64	746.9	67.0	4.7	1.7	9.5	15.2	58.0	34.2	28.4
9	x	x			20.06	16.29	64	814.3	110.4	-	-	188.7	14.2	-	-	228.5
10	x	x		x	20.31	16.05	63	849.7	44.1	0.5	1.5	7.5	14.1	12.7	29.3	25.7
11	x	x	x		20.09	16.99	65	768.3	85.1	-	-	120.6	15.1	-	-	153.1
12	x	x	x	x	19.61	16.81	66	727.1	45.1	0.7	1.4	6.0	15	12.7	24.8	22.7
13	Monolithic				24.39	14.55	49	1499.2		-	-	-	-	-	-	-

The most important component of the algorithm is the LP-based cut. Without this cut, several solutions of the MP are wrongly evaluated. Without the four components (first line of Table 10), the algorithm explored on average 16,211 nodes, generating 44,506 cuts which cut out all but 16 nodes, on average. The high number of explored nodes used most of the solution time to compute the SPs (2,733.5 seconds on average). Moreover, the quality of the solutions is very poor, only one instance from the dataset was solved in the time limit of 3,600 seconds. The partial cuts compensate part of the information when

the LP-cuts are not used (second line). Most of the overestimated solutions from the master problem are corrected by solving the sequencing of the individual stations, generating on average 23,087 L1 cuts. These cuts reduce drastically the number of explored nodes (from 16,237 to 1,441 on average), allowing more time for the algorithm for the exploration of the MP. The preprocessing in the absence of the LP-based cuts (lines 3 and 4) presents only limited improvement.

The Benders' Decomposition with the LP-based Cut in any configuration (lines 5 to 12) outperforms the Monolithic model (line 13). Although each algorithm feature affects the search of the cut generation procedure, the results measured in average solution quality or number of solved solutions just slightly change. Both the preprocessing and the idle-time-based cut improve the information of the MP, so that fewer nodes are needed to be solved as SPs. Without the partial cuts, the number of explored nodes reduces from 264.2 (line 5) to 202.9 and 203.7 by adding Idle-Time cuts and preprocessing, respectively, or to 135.7 by adding both. The exploration of fewer nodes frees time to the search of MP. The reduction of explored nodes, however, does not directly translate to better solutions or more instances solved within the time limit.

Line 12 of the table contains the results of the complete algorithm. Using all features, the algorithm obtains on average marginally better solutions using less time. How fast the MP's solutions are found plays a significant role in the algorithm, because with all improvements only few nodes need the sequencing to be solved. Therefore, the variation of solution times of the search procedure for the MP based on a commercial solver is stronger than the effect of the features in the algorithm.

The detailed results for the algorithm with all components are shown in Table 11. The symbols representing the columns are the same as in Table 10. The instances are categorized based on their parameters (OS, PL, and LM), which allows the comparison of solution difficulty and quality between different parameter constellations. These parameters can be interpreted as degrees of flexibility in the problem. The ordering strength (OS) measures how the assignments are restricted. A low value has fewer precedence relations and therefore more flexibility to assign the tasks along the stations. This effect of OS can be clearly seen in the upper bound of the solutions. The higher the OS is, the larger is the expected utility work keeping the other parameters constant. The effect of OS in the solution time depends on the other factors. For a high OS (0.9), the limited assignment possibilities yield in instances that are solved quickly. Low values of OS (0.2) are mostly solved easily for low PL or high LM, because solutions with no utility work exist and are therefore optimal. The tightest condition (PL = 0.95 and LM = 1.2), however, produce the most difficult instances to solve. None of the 5 instances with these parameters were solved, and the lower bound after 3,600 seconds remains at zero. The instances with average assignment flexibility (OS = 0.6) are, on average, the ones that take more time to solve. For the highly loaded instances (PL = 0.95), there are instances that are not solved to optimality for every value of the length multiplier. The second impacting factor is the average processing load (PL). It measures how loaded on average the stations are. Instances with low PL (0.90) are, on average, easier to solve and require

much less utility work than the ones with high PL (0.95). For most of the instances with $PL = 0.90$ and $OS \leq 0.6$, the optimal solutions exhibit no utility work. Finally, the station length multiplier (LM) corresponds to the sequencing flexibility. High LM values result in less utility work at the cost of longer stations. The increase of station length reduces the expected utility work considerably from LM = 1.2 to 1.5, at least for the instances with OS = 0.6. A further increase to LM = 2.0 results in a smaller decrease on the expected utility work. It is interesting to observe that even for values of LM = 2.0, instances with restricted assignment possibilities (OS = 0.9) still need a significant amount of utility work. The combination OS = 0.9 and LM = 1.2 is not even considered in the dataset, since all generated instances for this constellation were proven infeasible.

Table 11: Results of the algorithm with all its components for the groups of different instances. Each line contains the average results of 5 instances of each combination of ordering strength (OS), average processing load (PL), and station length multiplier (LM).

Parameters						Sol. time		# Nodes				# Cuts		
OS	PL	LM	UB	LB	Opt	Master	Sub	L1	L2&3	Full	Inc	L1	L2&3	Full
0.2	0.90	1.2	0	0	5	60.5	51.9	1.4	4.2	8.4	13.6	23.2	25.4	35.8
0.2	0.90	1.5	0	0	5	1.18	14.6	0.2	0.2	0.8	12.4	3.6	3.8	10.0
0.2	0.90	2.0	0	0	5	1.16	1.92	0	0	0	11.6	0	0	0
0.2	0.95	1.2	33.88	0	0	3441.2	158.9	1.75	4.25	15.75	11.5	68	76.8	71.0
0.2	0.95	1.5	0	0	5	560.6	38.7	3.4	5.2	9.2	22.8	16.6	20.2	28.8
0.2	0.95	2.0	0	0	5	17.1	3.26	0	0	0	17.4	0	0	0
0.6	0.90	1.2	1.15	0	3	1627.9	89.3	0.8	1.8	16.4	12.4	30.0	25.8	57.4
0.6	0.90	1.5	0	0	5	3.08	11.3	0.4	0.6	2.6	14.6	4.4	6.4	9.8
0.6	0.90	2.0	0	0	5	1.92	2.42	0	0	0	14.2	0	0	0
0.6	0.95	1.2	81.09	34.66	2	2853.4	223.3	0.6	1.4	16.8	8.4	56.6	68.2	77.2
0.6	0.95	1.5	7.01	2.60	2	2091.2	106.6	2.0	5.4	26.8	23.4	10.2	36.2	69.2
0.6	0.95	2.0	3.07	2.37	4	923.8	4.26	0	0	0	20	0	0	0
0.9	0.90	1.5	9.00	9.00	5	2.56	12.2	0	0.4	1.4	13.0	2.0	1.8	4.6
0.9	0.90	2.0	6.23	6.23	5	4.32	3.06	0	0	0	16.4	0	0	0
0.9	0.95	1.5	113.39	113.39	5	2.88	28.0	0.2	0	1.4	11.0	1.4	1.6	14.0
0.9	0.95	2.0	100.67	100.67	5	8.68	3.8	0	0	0	19.6	0	0	0

The dataset is also solved with a time limit of 14,400 seconds. With this time limit, 73 out of the 80 instances could be solved to optimality. All results can be found in the online supporting information material.

The convergence behavior of the algorithm is exemplified for instance number 22 in Fig. 4. The instance is chosen for the graphical example because it is solved within one hour and presents several solutions tested until the convergence. The plot shows all evaluated assignments between 30 seconds of computation until the convergence at 2203 seconds. The dark points represent feasible solutions and their objective function evaluations on the master problem. The light points show the objective value of the solution (aligned vertically) after the solution of all or some of the subproblems. The shaded region shows the best objective function value obtained until that point in time. Note that the master problem

only presents assignments to be tested that are bellow the 'best solution found' value. After solving the subproblems, the solution is either discarded by adding cuts or the best solution is updated.

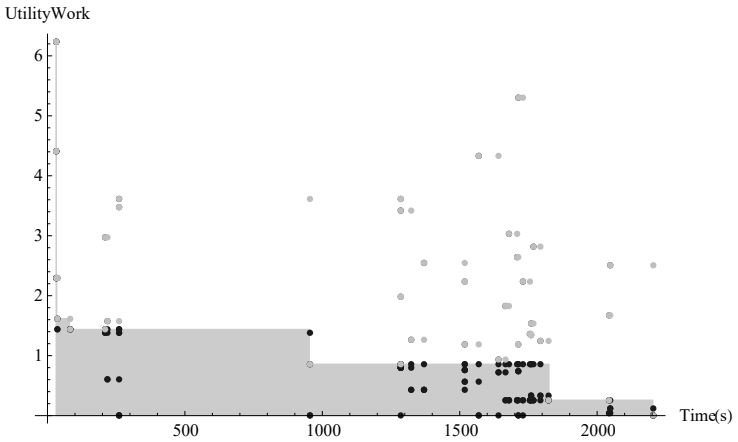

Figure 4: Plot of the convergence behavior of the algorithm for instance number 22. The plot shows the solutions found after 30 seconds of running time until convergence. Dark points are solution candidates found by the master problem, while lighter points show their objective function evaluation using the subproblems. The shaded area shows the best solution found until the time given.

5.4. Evaluation of the stochastic solutions on larger instances under real-world inspired demand scenarios.

In this section the solution of the second dataset is used to evaluate the value of treating stochastic demand scenarios. The 25 instances are solved with the proposed algorithm, as well as the 'Wait-and-See' solutions and the solution for the average demand. The 'Wait-and-See' solutions are obtained by solving the balancing and sequencing problem separated for each demand scenario. This solution considers that data is deterministic and known in the plan phase. According to Birge & Louveaux (2011), the 'Wait-and-See' solution is a lower bound for the problem (in minimization problems), while the solution for the average scenario is an upper bound.

The results of all instances are summarized in Table 12. The balancing based on the average demand scenario is given in the second and third columns. The instances are solved for one demand scenario considering the demand [7, 4, 2, 2, 2], which approximates the empirical annual average. The result of the average instance is given in the 'Exp.' column, as an expected value of the approach. The column 'Realized' contains the evaluation of the average balancing solution under all the 12 monthly scenarios. The average solution should be a lower bound of the realized value. A very small deviation of 0.01 is observed in 'Jaeschke' instance, that occurs because the discrete vector used for the average demand is not exactly equal to the expected demand. The difference is however small and can be neglected for the comparison. Only one instance is not solved to optimality within 43,200 seconds ('Barthol2'), for which the lower and upper bounds are given. The next two columns contain the solution value of the stochastic solution method and the time used for the computations. A time limit of 43,200 seconds is used. For the

instances not solved to optimality, the lower and upper bounds are given as intervals. The Wait-and-See solutions are also given within bounds along with the solution time. For this approach, a limit of 43,200 seconds is given for each individual demand scenario. Some instances exceeded the RAM capacity of the computer used within the 43,200 seconds, so that the results marked with † are obtained with less than the time limit, before the RAM overload occurred.

Table 12: Results for the second dataset. The intervals [LB; UB] contain the lower and upper bound of the objective values of the average demand approach the stochastic approach, and the wait-and-see approach.

	Average		Stochastic		Wait-and-See		Stochastic value	
Instance	Expected	Realized	[LB; UB]	Time	[LB; UB]	Time	VSS	EVPI
Arcus1	0	364.77	[216.09; 273.86]	43200	[10.47; 12.46]	130404	[90.9; 148.68]	[203.63; 263.39]
Arcus2	0	515.13	[18.54; 189.65]	43200	[2.65; 2.81]	43647	[325.48; 496.59]	[15.73; 187.00]
Barthold	0	14.2	0	65.32	0	11.8	14.2	0
Barthol2	[0; 11.18]	29.35	[0; 90.71]	43200	[0; 2.67]†	66909†	[0; 29.35]*	[0; 90.71]*
Bowman	4.12	4.11	4.11	1.76	0.77	0.61	0	3.34
Buxey	8.88	10.2	10.2	19.03	1.93	70.06	0	8.27
Gunther	6.76	8.39	8.38	472.73	1.3	133.44	0.01	7.09
Hahn	264.62	308.75	301.51	7.15	48.8	8.7	7.24	252.71
Heskiaoff	0	2.16	0	2.54	0	0.65	2.16	0
Jackson	0	0.03	0.01	2.16	0.001	0.31	0.02	0.01
Jaeschke	0.47	0.46	0.46	0.1	0.11	0.19	0	0.36
Kilbridge	0	1.27	0.11	2909.25	[0; 0.002]	43229	1.16	[0.113; 0.115]
Lutz1	84.68	151.76	151.76	66.49	25.96	16.45	0	125.8
Lutz2	3.85	5.77	5.43	12874	0.69	10564	0.34	4.74
Lutz3	0	4.79	1.86	19156	0.07	932.89	2.92	1.79
Mansoor	3.18	3.68	3.68	0.63	0.49	0.79	0	3.19
Mertens	1.71	1.77	1.77	0.08	0.32	0.07	0	1.45
Mitchell	0	0.1	0.02	1.94	0.002	1.53	0.07	0.02
Mukherje	0	21.86	19.09	31932.71	[1.333; 1.334]	43241	2.77	[17.756; 17.757]
Roszieg	0	0.29	0.25	7.45	0.014	5.07	0.04	0.236
Sawyer	12.59	13.53	13.53	24.05	2.24	64.11	0	11.11
Scholl	0	289.12	[13.25; 394.25]†	5402.6†	[1.11; 7.51]	131967	[0; 275.87]*	[5.74; 393.15]
Tonge	0	11.76	[7.17; 8.1]	43200	[0.415; 0.435]	45472	[3.66; 4.58]	[6.73; 7.68]
Warnecke	0.35	6.61	[0; 2.81]	43200	[0; 0.040]	146173	[3.81; 6.61]	[0; 2.81]
Wee-Mag	0	5.00	[0; 1.45]	43200	[0; 0.01]	87362	[3.55; 5.00]	[0; 1.45]

The last two columns of Table 12 contain the well-known stochastic measures 'Value of Stochastic Solution' (VSS) and the Expected Value of Perfect Information' (EVPI) (Birge & Louveaux, 2011). The VSS measure is the difference of the objective value of the solution based on the average scenario and the stochastic solution. This value shows how important the consideration of stochastic demand is. The EVPI is the difference between the stochastic solution and the Wait-and-See solution. This value shows the impact of stochasticity in the solution comparing to the model with all information. For the solutions that are not solved to optimality, lower and upper bounds for the stochastic measure are given. For the instances of 'Barthol2' and 'Scholl', the upper bound of the stochastic solution is larger than the objective function of the average solution. Therefore, the 'VSS' must lay between 0 and the difference of

the objective value of the average solution and the lower bound of stochastic solution.

In this dataset, the realized solution when the problem is solved for the average demand scenario is different to the wait-and-see solution for every instance. This difference shows that the stochasticity in demand has a strong effect on the performance of an assembly line. The stochastic approach considers all scenarios and solves the balancing problem for the minimal expected utility work for a set of possible scenarios. Considerable relative reductions can be observed between the average and the stochastic approach for the at least half of the presented instances (Arcus1, Arcus2, Barthold, Heskiaoff, Jackson, Kilbridge, Lutz3, Mitchell, Mukherje, Tonge, Warnecke, Wee-Mag).

The obtained Expected Values of Perfect Information (EVPI) show that a large proportion of the uncertainty cannot be accounted for previously. In only 8 instances the EVPI is provenly larger than the VSS, a case in which more than half of the uncertainty effects would be hedged by the stochastic approach. It is interesting to notice that the EVPI of two instances is zero (Barthold and Heskiaoff), while in 7 instances the average solution and the stochastic solution tied.

5.5. Effect of the number of scenarios

In this section, a test to determine the effect of the number of demand scenarios is performed. The instance Hahn' is selected for the comparisons. For this test, more demand scenarios are drawn randomly based on the models' proportion of the annual sales. Instances from 10 to 50 scenarios are proposed with MPSs of 10 and 15 products. In total, 10 instances are proposed and solved to optimality.

Table 13 contains a summary of the results with the objective function of the average demand scenario approach (also [7, 4, 2, 2, 2]), the stochastic approach minimizing the expected utility work, and the balancing optimization of every individual scenario (wait-and-see). The value of stochastic solution (VSS), expected value of perfect information (EVPI) are also given. As for the other datasets, the instances and all solution files are found in the online supplementary material.

Table 13: Results for the third dataset. 'Heskiaoff' instance is solved for 10 to 50 demand scenarios and an MPS with 10 or 15 products.

Instance	Average		Stochastic		Wait-and-See		Stochastic value		
(Scen.) - (MPS)	Expected	Realized	Opt Val.	Time	Opt Val.	Time	VSS	EVPI	$\frac{\text{VSS}}{\text{VSS+EVPI}}$
Hahn - 10 - 10	439.79	579.75	570.41	4.2	57.04	1.7	9.34	513.37	1.8%
Hahn - 20 - 10	25.82	555.42	490.19	43.7	20.39	47	65.23	469.8	12.2%
Hahn - 30 - 10	393.41	541.38	506.96	36.9	15.74	12.3	34.42	491.22	6.6%
Hahn - 40 - 10	294.68	502.78	493.97	215.3	11.25	70.6	8.81	482.72	1.8%
Hahn - 50 - 10	126.68	409.12	340.07	631.9	5.86	67.8	69.05	334.21	17.1%
Hahn - 10 - 15	208.79	424.85	401.78	122.3	36.05	1312.5	23.07	365.73	5.9%
Hahn - 20 - 15	42.18	407.89	317.99	10010.6	14.8	7937.2	89.9	303.2	22.9%
Hahn - 30 - 15	135.59	446.98	374.96	2409.3	11.56	13062.4	72.02	363.4	16.5%
Hahn - 40 - 15	202.35	447.35	388.71	4553.1	8.92	21765.5	58.64	379.79	13.4%
Hahn - 50 - 15	296.82	402.24	398.34	2935.3	7.32	46963.1	3.90	391.02	1.0%

From Table 13, it can be observed that the solution times increase based on the number of scenarios

and the length of the MPS. The most significant increase is due to the MPS, once the subproblems take more time to be solved. Therefore, the proposed Benders Decomposition should be best applied to scenarios with limited MPS length. The number of scenarios greatly increase the solution time for the wait-and-see method, since more balancing problems have to be solved. That is not the case for the stochastic approach, which only has one balancing solution for all scenarios. For the long MPS, the stochastic approach solves the instance in significantly less time for 4 out of 5 instances. For the MPS of length 15, no increase of solution time is observable for increments of the number of demand scenario.

In respect to the expected utility work, the line costs decline slightly with more demand scenarios. This effect can be clearly seen in the wait-and-see case, while they are less apparent for the average and stochastic scenarios.

Along with VSS and EVPI, Table 13 shows the value of $\frac{\text{VSS}}{\text{VSS+EVPI}}$. This value represents the ratio between the difference of the stochastic and wait-and-see solution to the average and wait-and-see solution. In other words, this ratio is the proportion of the uncertainty that can be hedged by the stochastic approach in respect to the average solution and the wait-and-see solutions. A ratio of 0 represents no improvement from the stochastic solution in respect to the average solution. A ratio of 1 means that the stochastic solution hedges all uncertainties and is cost equivalent to the wait-and-see approach. The ratio is expected to assume intermediary values and varies between 1.0% to 22.9% for the tested instances. These values show that the wait-and-see solution quality is by far not achievable, but small or medium improvements are provided by solving the stochastic version of the problem. The total cost reduction for the 10 instances is 9.2%, which is very relevant for the operations costs of assembly lines.

5.6. Discussion and managerial insights

In order to test the algorithm and evaluate the uncertainty in the demand of mixed-model assembly lines, three datasets are proposed and tested. From the first dataset, instances with the same number of tasks and stations are used to evaluate the effect of the tasks' and line's characteristics. As shown in Table 11, the amount of expected utility work is strongly correlated to the order strength (OS), the average load of the assembly line (PL), and the station length (LM). Lower OS provide few precedence restrictions, so that several assignment combinations are feasible and the expected utility work is expected to be smaller than otherwise. For the cases with OS > 0.6, an average expected load of 90% for the whole line provided solutions with low (or no) utility work. Increasing the length of the stations showed to reduce significantly for short stations (120% of the cycle time to 150%) but rather limit, since further increases provide fewer incremental advantages. Therefore, from the insights of the first dataset, the effects of both increasing the cycle time and elongating stations can be used to provide low expected utility work solutions. Although the quality of solutions strongly depend on the instance, values of 95% seem to be adequate to low OS, while further reductions (90%) may be necessary for stricter assignments. For the station length, 120% or 150% of the cycle time are recommended based on the dataset.

The second dataset uses demand scenarios based on a year data of vehicle's licensing for a manufacturing site. For these instances, the Value of the Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI) are given. The results show that the stochasticity effect depends strongly on the instance, ranging from no difference as in 'Barthold' to several hundred cost units as in 'Hahn'. The results show that the benefit of dealing with the stochasticity significantly reduce the expected utility work, but only part of the uncertainty can be hedged by the stochastic solution. The dataset also showed that large instances with more than 100 tasks can also be solved to optimality (as 'Barthold'), although the solution times are larger than the ones for the average case. For the largest instance 'Scholl', for instance, the stochastic approach was unable to deliver a better solution comparing to the average demand approach given RAM and time limitations. Therefore, the proposed method is recommended to small and medium instances of the problem.

A third dataset is used to determine the effect of the number of demand scenarios and the length of the MPS. As observable in Table 13, the solution approach is much more sensible to the length of the MPS than the number of demand scenarios. Furthermore, the stochastic approach is less sensible to the number of demand scenarios than solving all balancing problems for each demand scenario individually for long MPSs. The tests showed that the improvement of solving the balancing problem including the demand uncertainty reduce the expected utility work between 1.0% to 22.0% (with an average of 9.2%). As no clear trend between the expected utility work and the number of scenarios is found, the stochastic approach is indicated whenever demand presents uncertainties.

Although the parametric and real-world inspired datasets are useful to test and justify the use of the algorithm, further works are necessary for the application in a real-world scenario. The main issues with a realistic scenario generation do respect to the time frame, amount of information, and the ability to foresee the future demand. The assignment of tasks and equipment among an assembly line is a medium to long time decision, so that planning periods of several months to three years should be considered. An assembly line is generally adapted or rebalanced periodically to cope with new products and changes on production levels. Therefore, the definition of planning horizon is decisive for the source of uncertainty dealt with. Shorter planning periods may be adequate to consider daily or weekly demand variations for the interval between two rebalancing opportunities. Longer planning periods may cover seasonal variations of demand and the integration or exclusion of models. The long lasting alternative may require data from several years of historical data.

A second issue on generating demand scenarios is to use the historical information to predict the future demand. For short planning horizons when no new model is introduced or removed from the product pool, most of the demand variation could be attributed to the product sale values. Therefore, representative daily or weekly demand scenarios of the past may be an adequate input for the optimization model.

For longer planning horizons, the demand variation cannot be only explained with random variations of sale values. Even with the access of years of historical data, the future sales are dependent of the

market, competitors, economy, etc. One alternative in this case is to first build a prediction model for the future demand and use a set of scenarios as input of the optimization problem. Forecast demand is a research field on its own. One example is given by Fantazzini & Toktamysova (2015), who forecast demand of German manufactures using multivariate models.

6. Conclusion

In this paper, an exact method for the solution of integrated balancing and sequencing of mixed-model assembly lines is presented. The approach unifies two problems with different time scales: the long-term balancing and the short-term sequencing problem. The demand is considered to be stochastic and is modeled through discrete scenarios. The objective of the approach is to find the task assignments that have the lowest average expected utility work for the given demand scenarios.

The solution method is based on the Benders' decomposition (Benders, 1962) and decomposes the formulation in balancing and sequencing problems. This way, the sequencing problems can be solved independently, exploiting the structure of the stochastic problem. As the considered subproblem contains binary variables, a combinatorial adaptation of the algorithm is used. Valid inequalities and preprocessing are presented to tighten the formulation. It is shown that the cut based on the linear relaxation of the SP can be derived in a single expression. Furthermore, a further decomposition of the subproblem (partial decomposition) producing lower bounds on the objective function is used to generate low-density cuts for the master problem.

Three datasets are proposed to test the algorithm and the effect of the stochasticity in the objective function. A first dataset consisting of 80 medium-sized instances (50 tasks) is proposed from which 66 were solved to optimality within 3600 seconds. The instances are built considering minimal part sets (MPSs) from up to 10 product models. As the considering sequencing model is cyclic, multiples of the MPS are easily replicable and can be - to some extent - combined to generate larger sequences.

For the proposed first dataset, the flexibility of task assignments plays an important role in the necessity of utility work. Assembly lines with few precedence relations can mostly be balanced with an occupancy of 95% without utility work in the dataset. The results also show a significant utility work reduction from a station length of 150% of the cycle time comparing to 120% and a smaller effect for further increases on length. Algorithmically, the instances with high workload, low stations size, and high assignment flexibility are the most difficult ones for the proposed algorithm.

In the second and third dataset, the demand scenarios are inspired in real world data. The average demand and every individual scenario are also solved for these datasets to determine the value of stochastic solution (VSS) and the expected value of perfect information (EVPI). The tests show that hedging for demand uncertainty can significantly reduced the expected utility work, although the improvement is instance dependent.

The assembly line balancing is mostly modeled as a deterministic problem considering all data to be

known in the planning phase. In practice, not only the data may be uncertain, but also the production mix is dynamic and changes based on sales. By considering multiple demand scenarios, the balancing solution may be more flexible to production rate fluctuations and requires fewer modifications throughout the production life of an assembly line.

As future work, the research on assembly line balancing and the real-world lines can be bridged by better solution methods to solve larger instances dealing with more uncertainties and by considering practical restrictions. The product models, for instance, may not be freely sequenced for the final assembly, but may also consider the output of other parts of the plant. Another research direction is the combination of the solution method with demand forecasting techniques and statistical models. Furthermore, planned rebalancing phases between different monthly production plans are also possible, so that a rebalancing can also be used to hedge against demand uncertainty.

References

Akpinar, S., Elmi, A., & Bektaş, T. (2017). Combinatorial Benders cuts for assembly line balancing problems with setups. *European Journal of Operational Research*, *259*, 527–537. doi:10.1016/j.ejor.2016.11.001.

ANFAVEA, A. N. d. F. d. V. A. (2019). *Autoveículos - Produção, licenciamento, exportações em unidades de montados e CKD (desmontados), exportações em valor e emprego*. Technical Report ANFAVEA. URL: http://www.anfavea.com.br/estatisticas.

Bard, J. F., Dar-Elj, E., & Shtub, A. (1992). An analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research*, *30*, 35–48. doi:10.1080/00207549208942876.

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, *142*, 259–277. doi:10.1016/j.ijpe.2012.10.020.

Benders, J. F. (1962). Partitioning Procedures for Solving Mixed-variables Programming Problems. *Numer. Math.*, *4*, 238–252. doi:10.1007/BF01386316.

Bentaha, M. L., Battaïa, O., & Dolgui, A. (2014). A sample average approximation method for disassembly line balancing problem under uncertainty. *Computers and Operations Research*, *51*, 111–122. URL: http://dx.doi.org/10.1016/j.cor.2014.05.006. doi:10.1016/j.cor.2014.05.006.

Bentaha, M. L., Dolgui, A., & Battaïa, O. (2015). A bibliographic review of production line design and balancing under uncertainty. *IFAC-PapersOnLine*, *28*, 70–75. doi:10.1016/j.ifacol.2015.06.060.

Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*. (2nd ed.). New York: Springer-Verlag. doi:10.1007/978-1-4614-0237-4.

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.

Boysen, N., Fliedner, M., & Scholl, A. (2009a). Production planning of mixed-model assembly lines: overview and extensions. *Production Planning & Control*, *20*, 455–471. doi:10.1080/09537280903011626.

Boysen, N., Fliedner, M., & Scholl, A. (2009b). Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research*, *192*, 349–373. doi:10.1016/j.ejor.2007.09.013.

Bukchin, J., Dar-El, E. M., & Rubinovitz, J. (2002). Mixed model assembly line design in a make-to-order environment. *Computers & Industrial Engineering*, *41*, 405–421. doi:10.1016/S0360-8352(01)00065-1.

Chica, M., Bautista, J., & de Armas, J. (2018). Benefits of robust multiobjective optimization for flexible automotive assembly line balancing. *Flexible Services and Manufacturing Journal*, (pp. 1–29). doi:10.1007/s10696-018-9309-y.

Chica, M., Bautista, J., Cordón, Ó., & Damas, S. (2016). A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega*, *58*, 55–68. doi:10.1016/j.omega.2015.04.003.

Chica, M., Cordón, Ó., Damas, S., & Bautista, J. (2013). A robustness information and visualization model for time and space assembly line balancing under uncertain demand. *International Journal of Production Economics*, *145*, 761–772. doi:https://doi.org/10.1016/j.ijpe.2013.05.030.

Codato, G., & Fischetti, M. (2006). Combinatorial Benders' Cuts for Mixed-Integer Linear Programming. *Operations Research*, *54*, 756–766. doi:10.1287/opre.1060.0286.

Costa, A. M., Cordeau, J.-F., Gendron, B., & Laporte, G. (2012). Accelerating benders decomposition with heuristic master problem solutions. *Pesquisa Operacional*, *32*, 3–19. doi:10.1590/S0101-74382012005000005.

Emde, S., & Boysen, N. (2012). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *European Journal of Operational Research*, *217*, 287–299. URL: http://dx.doi.org/10.1016/j.ejor.2011.09.013. doi:10.1016/j.ejor.2011.09.013.

Emde, S., & Gendreau, M. (2017). Scheduling in-house transport vehicles to feed parts to automotive assembly lines. *European Journal of Operational Research*, *260*, 255–267. doi:10.1016/j.ejor.2016.12.012.

Fantazzini, D., & Toktamysova, Z. (2015). Forecasting German car sales using Google data and multivariate models. *International Journal of Production Economics*, *170*, 97–135. doi:https://doi.org/10.1016/j.ijpe.2015.09.010.

FENAFABRE, F. N. d. D. d. V. A. (2019). *Dados de mercado FENAFABRE. Informativo - Emplacamentos*. Technical Report FENAFABRE. URL: http://www.fenabrave.org.br/Portal/conteudo/emplacamentos.

Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Benders decomposition without separability: A computational

study for capacitated facility location problems. *European Journal of Operational Research*, *253*, 557–569. doi:10.1016/j.ejor.2016.03.002.

Fischetti, M., Ljubić, I., & Sinnl, M. (2017). Redesigning Benders Decomposition for Large-Scale Facility Location. *Management Science*, *63*, 2146–2162. doi:10.1287/mnsc.2016.2461.

Hop, N. V. (2006). A heuristic solution for fuzzy mixed-model line balancing problem. *European Journal of Operational Research*, *168*, 798–810. doi:10.1016/j.ejor.2004.07.029.

Kao, E. P. C. (1976). A Preference Order Dynamic Program for Stochastic Assembly Line Balancing. *Management Science*, *22*, 1097–1104. doi:10.1287/opre.26.6.1033.

Karabati, S., & Sayın, S. (2003). Assembly line balancing in a mixed-model sequencing environment with synchronous transfers. *European Journal of Operational Research*, *149*, 417–429. doi:10.1016/S0377-2217(02)00764-6.

Kottas, J. F., & Lau, H. S. (1973). A cost-oriented approach to stochastic line balancing. *AIIE Transactions*, *5*, 164–171. doi:10.1080/05695557308974897.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, *13*, 133–142. doi:10.1016/0167-6377(93)90002-X.

Li, J., & Gao, J. (2014). Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research*, *52*, 3552–3567. doi:10.1080/00207543.2013.874603.

Lopes, T. C., Michels, A. S., Sikora, C. G. S., & Magatão, L. (2019). Balancing and cyclical scheduling of asynchronous mixed-model assembly lines with parallel stations. *Journal of Manufacturing Systems*, *50*, 193–200. doi:10.1016/j.jmsy.2019.01.001.

Lopes, T. C., Michels, A. S., Sikora, C. G. S., Molina, R. G., & Magatão, L. (2018). Balancing and cyclically sequencing synchronous, asynchronous, and hybrid unpaced assembly lines. *International Journal of Production Economics*, *203*, 216–224. doi:10.1016/j.ijpe.2018.06.012.

Lopes, T. C., Sikora, C. G. S., Michels, A. S., & Magatão, L. (2020). Mixed-model assembly lines balancing with given buffers and product sequence: model, formulation comparisons, and case study. *Annals of Operations Research*, *286*, 475–500. doi:10.1007/s10479-017-2711-0.

Manavizadeh, N., Rabbani, M., Moshtaghi, D., & Jolai, F. (2012). Mixed-model assembly line balancing in the make-to-order and stochastic environment using multi-objective evolutionary algorithms. *Expert Systems with Applications*, *39*, 12026–12031. doi:10.1016/j.eswa.2012.03.044.

McCormick, S., & Rao, U. (1994). Some complexity results in cyclic scheduling. *Mathematical and Computer Modelling*, *20*, 107–122. doi:10.1016/0895-7177(94)90210-0.

McCormick, S. T., Pinedo, M. L., Shenker, S., & Wolf, B. (1989). Sequencing in an Assembly Line with Blocking to Minimize Cycle Time. *Operations Research*, *37*, 925–935. doi:10.1287/opre.37.6.925.

McMullen, P. R., & Frazier, G. V. (1997). A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics*,

51, 177–190. doi:10.1016/s0925-5273(97)00048-0.

Michels, A. S., Lopes, T. C., & Magatão, L. (2020). An exact method with decomposition techniques and combinatorial Benders' cuts for the type-2 multi-manned assembly line balancing problem. *Operations Research Perspectives*, . doi:https://doi.org/10.1016/j.orp.2020.100163.

Michels, A. S., Lopes, T. C., Sikora, C. G. S., & Magatão, L. (2018). The Robotic Assembly Line Design (RALD) problem: Model and case studies with practical extensions. *Computers and Industrial Engineering*, *120*, 320–333. doi:10.1016/j.cie.2018.04.010.

Michels, A. S., Lopes, T. C., Sikora, C. G. S., & Magatão, L. (2019). A Benders' decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem. *European Journal of Operational Research*, *278*, 796–808. doi:10.1016/j.ejor.2019.05.001.

Mosadegh, H., Fatemi Ghomi, S. M., & Süer, G. A. (2019). Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and Q-learning based simulated annealing hyper-heuristics. *European Journal of Operational Research*, . doi:10.1016/j.ejor.2019.09.021.

Mosadegh, H., Ghomi, S. M. T. F., & Süer, G. A. (2017). Heuristic approaches for mixed-model sequencing problem with stochastic processing times. *International Journal of Production Research*, *55*, 2857–2880. doi:10.1080/00207543.2016.1223897.

Mossa, G., Boenzi, F., Digiesi, S., Mummolo, G., & Romano, V. A. (2016). Productivity and ergonomic risk in human based production systems: A job-rotation scheduling model. *International Journal of Production Economics*, *171*, 471–477. doi:10.1016/j.ijpe.2015.06.017.

Oesterle, J., Amodeo, L., & Yalaoui, F. (2017). A comparative study of Multi-Objective Algorithms for the Assembly Line Balancing and Equipment Selection Problem under consideration of Product Design Alternatives. *Journal of Intelligent Manufacturing*, (pp. 1–26). doi:10.1007/s10845-017-1298-2.

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.

Özcan, U., Kellegöz, T., & Toklu, B. (2011). A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem. *International Journal of Production Research*, *49*, 1605–1626. doi:10.1080/00207541003690090.

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. doi:10.1016/j.jmsy.2015.05.004.

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.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*, 801–817. doi:10.1016/j.ejor.2016.12.005.

Reeve, N. R., & Thomas, W. H. (1973). Balancing stochastic assembly lines. *AIIE Transactions*, *5*, 223–229. doi:10.1080/05695557308974905.

Ritt, M., & Costa, A. M. (2018). Improved integer programming models for simple assembly line balancing and related problems. *International Transactions in Operational Research*, *25*, 1345–1359. doi:10.1111/itor.12206.

Ritt, M., Costa, A. M., & Miralles, C. (2016). The assembly line worker assignment and balancing problem with stochastic worker availability. *International Journal of Production Research*, *54*, 907–922. doi:10.1080/00207543.2015.1108534.

Sawik, T. (2012). Batch versus cyclic scheduling of flexible flow shops by mixed-integer programming. *International Journal of Production Research*, *50*, 5017–5034. doi:10.1080/00207543.2011.627388.

Scholl, A. (1993). *Data of Assembly Line Balancing Problems*. Technical Report Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL) Darmstadt.

Silverman, F. N., & Carter, J. C. (1986). Cost-Based Methodology for Stochastic Line Balancing With Intermittent Line Stoppages. *Management Science*, *32*, 455–463. doi:10.1287/mnsc.32.4.455.

Simaria, A. S., Zanella De Sá, M., & Vilarinho, P. M. (2009). Meeting demand variation using flexible U-shaped assembly lines. *International Journal of Production Research*, *47*, 3937–3955. doi:10.1080/00207540701871044.

Sphicas, G. P., & Silverman, F. N. (1976). Deterministic equivalents for stochastic assembly line balancing. *AIIE Transactions*, *8*, 280–282. doi:10.1080/05695557608975078.

Thomopoulos, N. T. (1967). Line Balancing-Sequencing for Mixed-Model Assembly. *Management Science*, *14*, 59–75. doi:10.1287/mnsc.14.2.B59.

Tiacci, L. (2015a). Coupling a genetic algorithm approach and a discrete event simulator to design mixed-model un-paced assembly lines with parallel workstations and stochastic task times. *International Journal of Production Economics*, *159*, 319–333. doi:10.1016/j.ijpe.2014.05.005.

Tiacci, L. (2015b). Simultaneous balancing and buffer allocation decisions for the design of mixed-model assembly lines with parallel workstations and stochastic task times. *International Journal of Production Economics*, *162*, 201–215. doi:10.1016/J.IJPE.2015.01.022.

Van Slyke, R., & Wets, R. (1969). L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics*, *17*, 638–663. doi:10.1137/0117061.

Vrat, P., & Virani, A. (1976). A cost model for optimal mix of balanced stochastic assembly line and the modular assembly system for a customer oriented production system. *International Journal of Production Research*, *14*, 445–463. doi:10.1080/00207547608956618.

Yang, C., & Gao, J. (2016). Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research*, *65*, 139–148. doi:10.1016/j.cor.2015.07.007.

Yano, C. A., & Rachamadugu, R. (1991). Sequencing to Minimize Work Overload in Assembly Lines with Product Options. *Management Science*, *37*, 572–586. doi:10.1287/mnsc.37.5.572.