



Exploring the application of machine learning to the assembly line feeding problem

Emilio Moretti¹ · Elena Tappia¹ · Veronique Limère^{2,3} · Marco Melacini¹

Received: 15 February 2021 / Revised: 14 June 2021 / Accepted: 28 June 2021 / Published online: 28 July 2021
© The Author(s) 2021, corrected publication 2021

Abstract

As a large number of companies are resorting to increased product variety and customization, a growing attention is being put on the design and management of part feeding systems. Recent works have proved the effectiveness of hybrid feeding policies, which consist in using multiple feeding policies in the same assembly system. In this context, the assembly line feeding problem (ALFP) refers to the selection of a suitable feeding policy for each part. In literature, the ALFP is addressed either by developing optimization models or by categorizing the parts and assigning these categories to policies based on some characteristics of both the parts and the assembly system. This paper presents a new approach for selecting a suitable feeding policy for each part, based on supervised machine learning. The developed approach is applied to an industrial case and its performance is compared with the one resulting from an optimization approach. The application to the industrial case allows deepening the existing trade-off between efficiency (i.e., amount of data to be collected and dedicated resources) and quality of the ALFP solution (i.e., closeness to the optimal solution), discussing the managerial implications of different ALFP solution approaches and showing the potential value stemming from machine learning application.

Keywords Assembly system · Part feeding · Factory logistics · Optimization · Machine learning · End-to-end learning

1 Introduction

Several manufacturing companies are resorting to increased product variety and customization in order to face competition (Wiengarten et al. 2017). While it helps meeting customer expectations, this strategy also results in a high number of parts to be handled in assembly systems and calls for the adoption of mixed-model assembly lines, where a single line can make several product models in an arbitrary mix (Golz et al. 2012; Faccio 2014). In this context, growing attention is being put on the design of part feeding systems, required to supply a wide variety of parts to assembly stations in time for the start of assembly operations, but without

piling up inventories on the shop floor (Sternatz 2015; Emde and Schneider 2018).

Different feeding policies can be evaluated when designing the system and the final choice results from a trade-off among different performance criteria. For instance, *kitting* policy entails supplying assembly stations with unit loads that contain only the parts needed for one or few finished products (e.g., Sali and Sahin 2016). The adoption of this policy is an effective way to reduce inventories on the shop floor and time wasted by assembly operators walking and searching for parts, compared to the case in which the unit loads received from suppliers are delivered to the stations (Limère et al. 2012). On the downside, kitting gives rise to an increased material handling effort, mainly due to picking activities (Hanson and Medbo 2019).

Some contributions in academic literature have introduced approaches to select the best policy for a given assembly system, starting with Bozer and McGinnis (1992). More recent works have shifted the focus to hybrid feeding policies, i.e., the use of multiple feeding policies in the same assembly system, showing improvements in the overall system performance (e.g., Limère et al. 2012; Usta et al. 2017). In this direction, the Assembly Line Feeding Problem

✉ Emilio Moretti
emilio.moretti@polimi.it

¹ Department of Management, Economics and Industrial Engineering, Politecnico Di Milano, Milano, Italy

² Department Business Informatics and Operations Management, Ghent University, Ghent, Belgium

³ Flanders Make, Oude Diestersebaan 133, 3920 Lommel, Belgium

(ALFP) has been defined as the “unambiguous assignment of every part to a single line feeding policy” (Schmid and Limère 2019).

So far, the ALFP has been addressed in literature either by developing optimization models (e.g., Caputo et al. 2018; Baller et al. 2020; Schmid et al. 2021) or by categorizing parts and assigning these categories to policies based on the value of some characteristics of both the parts and the assembly system (e.g., Caputo and Pelagagge 2011; Usta et al. 2017). Both the approaches present some drawbacks. For the first approach, they lie in the large amount of data and in the specific skills and software required to build and run the optimization models, which are often NP-hard (e.g., Faccio et al. 2018; Schmid et al. 2021). For the second approach, they lie in the quality of the achieved solution, that could be far from the optimal one (Sali and Sahin 2016).

Recently, a few studies have introduced approaches based on the combined use of optimization and machine learning (ML), committing to predict the optimal solution (Abbasi et al. 2020; Bengio et al. 2020) with a lower data collection effort compared to optimization approaches (Larsen et al. 2018). So far, combined optimization-ML approaches have been applied in the fields of energy systems (Fischetti and Fraccaro 2019) and transportation management (Larsen et al. 2018; Abbasi et al. 2020), but they raise the attention towards the opportunity to support data-driven decision making in many more fields. As regards production systems, the available literature includes several contributions in which ML techniques are applied (Kang et al. 2020; Bertolini et al. 2021), but never in combination with optimization. However, a number of issues should be better tackled in order to foster further research on combined optimization-ML approaches. The solution quality compared to optimization has been evaluated only by Abbasi et al. (2020). Moreover, the feasibility of the solution predicted through the ML model is an open challenge in the research on combined optimization-ML approaches (Bengio et al. 2020), as well as the required data collection effort (Larsen et al. 2018). Furthermore, previous contributions focus on specific problems, neither formalizing the steps required for the implementation of such methods nor offering any general methodological guidelines.

Given these premises, this paper explores the application of ML to deal with the ALFP, aiming at improving the trade-off between efficiency (i.e., amount of data and resources needed to solve the ALFP) and solution quality (i.e., closeness to the optimal solution) of the approaches currently available in literature. First, we present a combined optimization-ML approach leveraging the *end-to-end learning* method (Bengio et al. 2020), according to which an optimization model is applied only to a sample of parts and its output is used to train a supervised ML model. The ML model training leads to the identification of a few relevant

factors explaining the optimal ALFP solution and a set of simple rules for assigning new parts to feeding policies without any expertise in the optimization and ML fields. Secondly, we test the proposed approach using data from an industrial case, so as to show how this approach performs in comparison to optimization in terms of solution feasibility, solution quality, and data collection effort. Therefore, the paper has a two-fold contribution. On one hand, it enriches the previous literature on both part feeding policies selection and ML approaches applied to production systems. On the other hand, it investigates and discusses whether the proposed approach represents an effective method to select feeding policies in contexts where data availability is critical or data collection is very time-consuming.

The remainder of the paper is organized as follows: Sect. 2 reports a literature review on the approaches to solve the ALFP as well as on previous applications of the end-to-end learning method. Section 3 describes the proposed approach, while Sect. 4 presents its application. Section 5 shows the comparison between our approach and those existing in literature and discusses the related managerial implications. Finally, Sect. 6 includes conclusions and future research directions.

2 Literature review

2.1 Approaches to solve the ALFP

A wide variety of terms have been used in literature to refer to alternative part feeding policies. Schmid and Limère (2019) formally define the ALFP and identify five main types of policies, namely line stocking, boxed-supply, sequencing, stationary kitting, and travelling kitting. Over the years, several approaches have been proposed in literature to solve the ALFP choosing among these policies. As shown in Fig. 1, all the approaches revolve around two main elements: performance measures to evaluate the effect of the ALFP solution on system performance, and factors that affect the policy selection. We classify them as *inductive* or *deductive*. Inductive approaches start from the definition of performance measures, based on which an optimization model is built to find the optimal ALFP solution; in a subsequent phase, a parametric analysis is carried out to assess the impact of some selected factors on the optimal solution. According to deductive approaches, instead, components are categorized into groups and each group is assigned to a feeding policy based on the value of one or multiple factors; afterwards, performance measures are computed by means of a descriptive model.

The performance measures considered in the reviewed literature (Table 1) refer to the overall cost or time related to part feeding operations. The overall cost is computed as



→ Inductive approach

←- Deductive approach

Fig. 1 ALFP solution approaches

the sum of different items, encompassing the costs directly accountable to part feeding processes (i.e. supermarket replenishment, unit load preparation, transportation from storage areas to assembly stations, and picking of parts at the stations) and the investments required for the part feeding equipment (e.g. Caputo et al. 2018), as well as the

opportunity cost of inventories (e.g. Sali and Sahin 2016) or occupied space (e.g. Usta et al. 2017) at the border of the assembly line. In some cases, also other costs are taken into account, including the stock-out cost (Faccio 2014) and the costs incurred in planning and control activities (Usta et al. 2017). The total handling time is given by the sum

Table 1 Performance measures considered in the reviewed literature

Reference by publication year	Inventory holding cost	Space occupation cost	Supermarket replenishment cost	Unit load preparation cost	Transport cost	Picking cost at the line	Investments	Other costs	Total handling time
Inductive approach									
Limère et al. (2012)			X	X	X	X			
Caputo et al. (2015)	X	X		X	X	X	X		
Limère et al. (2015)			X	X	X	X ^b			
Sali and Sahin (2016)	X			X	X	X ^b			
Caputo et al. (2018)	X	X	X ^a	X ^a	X ^a	X ^a	X		
Baller et al. (2020)			X	X	X	X ^b			
Schmid et al. (2021)			X	X	X	X ^b			
Deductive approach									
Battini et al. (2009)	X								X
Caputo and Pelagagge (2011)	X	X		X ^a	X ^a		X		
Faccio (2014)	X			X ^a	X ^a			X	
Usta et al. (2017)	X ^c	X	X	X	X			X	
Faccio et al. (2018)									X

^aAggregated into an overall personnel cost

^bConsidering variable operator walking distances based on the amount of stock at the border of the line

^cStorage cost at the central warehouse

of the handling time at the supermarket for picking and kit preparation activities, the internal transportation time, and the handling time at the assembly line (Battini et al. 2009; Faccio et al. 2018).

According to Fig. 1, Table 2 gives an overview of the factors considered in the reviewed literature, to perform the parametric analysis for inductive approaches or to select policies for deductive approaches. The part-related factors include size, weight, value, demand, and number of variants of a part; sometimes, also the Bill of Materials (BOM) coefficient, supply unit load type, and stations where the part is used are taken into consideration. The assembly-system related factors refer to characteristics of the finished products (e.g., average lot size, number of parts per product), of the assembly stations (e.g., number of stations in the line, storage space available at the border of the station), of the cost structure (e.g., picking efficiency, unit inventory holding cost), as well as other features such as the size of the unit loads delivered to assembly stations or the type of transport equipment.

Considering the inductive approaches, several optimization models have been proposed, differing by the objective function, constraints, and number of alternative feeding policies. For instance, the mixed integer programming model by Sali and Sahin (2016) allows to choose, for each part, between kitting, line stocking, and sequencing, and aims at cost minimization considering storage space constraints at the border of assembly stations, capacity constraints of trolley trains, takt time constraints as well as volume and weight limits for kits. The cost minimization model by Schmid et al. (2021), instead, takes into account all the five feeding policies and simultaneously addresses the ALFP and the allocation of each part to a storage location at the border of the assembly line; besides volume and weight constraints for kits, this model accounts for storage space constraints, allowing space borrowing among different assembly stations. Most of these models are NP-hard and cannot be solved in reasonable time with commercial integer programming solvers; for instance, Faccio (2014) points out that his model leads to a large-scale combinatorial optimization model which is hard to be solved in practice, while Schmid et al. (2021) show that their model can be solved efficiently only through a solving framework exploiting cuts.

As concerns the deductive approaches, different procedures have been introduced. For instance, the approach by Faccio et al. (2018) requires to both group parts into classes and assign each class to a feeding policy based on the value of the “part picking space index”, defined as the ratio between the number of parts per container and the container volume, multiplied by the daily demand for the part. The total handling time is computed afterwards through a descriptive model, in order to compare the ALFP solution with a scenario in which the same policy is used for all the parts. Another example is the approach presented by Usta

et al. (2017): parts are grouped into three classes through a hierarchical clustering technique considering the value of two factors (part size and occupied area) and then each cluster is associated to either line stocking or kitting according to qualitative considerations on the size and weight of parts and to the opinions of experts working in the company. Then, the activity-based costing approach is adopted to compute the overall costs stemming from the ALFP solution. In the reviewed approaches classified as deductive, the factors on which the ALFP solution is based are selected in a subjective way. The only exception is the one by Faccio (2014). In his approach, several scenarios are defined, differing in terms of BOMs, production mix, and picking productivity in the kit preparation activity and then, for each scenario, a simulation is carried out to identify, out of a discrete sample of randomly generated ALFP solutions, the one that leads to minimum costs; the two factors showing the highest statistical correlation with the minimum-cost solutions are selected, parts are clustered into nine groups through a Pareto ABC cross-matrix, and each group is assigned to a feeding policy. As shown by the authors, the two selected factors are not sufficient to clearly link all the part groups to a feeding policy, thus requiring the decision maker to rely on qualitative criteria. Moreover, the solutions found through simulation do not necessarily correspond to the optimal ones.

2.2 End-to-end learning method

The first applications of the end-to-end learning method can be traced back to the 1980s and 1990s, when neural networks were adopted to address different types of combinatorial optimization problems such as the travelling salesman problem (e.g. Hopfield and Tank 1985), the knapsack problems (e.g. Ohlsson et al. 1993), and the general assignment problems (e.g. Gong et al. 1995). These early applications were sometimes characterized by low reproducibility and their results were not satisfactory if compared to alternative heuristic approaches, especially considering the high CPU times due to the limited available computing power (Smith 1999). Recently, a renewal of interest in this research stream has been favored by the exponential growth in computing power and data availability. Recent contributions encompass both new attempts to leverage ML models to solve the travelling salesman problem (e.g., Vinyals et al. 2015; Bello et al. 2016) and successful applications of the end-to-end learning method in the fields of energy systems and transportation. For instance, Fischetti and Fraccaro (2019) propose to use a supervised ML model to estimate the optimal production value of offshore wind parks when selecting new sites. They show that their approach allows to speed up the solution process with respect to the use of an optimization model alone, but, at the same time, it provides better results than the qualitative approach usually applied by practitioners. Larsen et al.

Table 2 Factors considered in the reviewed literature

Reference by publication year	Part-related factors				Assembly system-related factors																			
	Size	Weight	Value (cost)	Demand	Demand variability	Number of variants	BOM coefficient	Supply unit load type	Finished products demand	Production mix variability	Number of product models	Average assembly lot size	Number of parts per finished product	Number of parts common to different product models	Number of stations used at a station	Storage space at each station	Picking productivity	Hourly labor cost	Unit space occupation cost	Unit inventory holding cost	Unit load size	Distance from central warehouse to assembly line	Type of transport equipment	
Inductive approach																								
Limère et al. (2012)	X					X		X											X					
Caputo et al. (2015)	X	X						X	X									X	X					
Limère et al. (2015)	X	X		X		X	X																	
Sali and Sahin (2016)						X										X						X		
Caputo et al. (2018)	X	X	X					X										X		X				
Schmid et al. (2021)	X	X		X		X			X					X		X								
Deductive approach																								
Battini et al. (2009)	X		X	X	X						X	X										X		
Caputo and Pelagaggi (2011)			X ^a																					
Faccio (2014)			X				X		X					X										
Usta et al. (2017)	X							X													X		X	
Faccio et al. (2018)							X ^a	X ^a														X ^a		

^aCombined in a single factor

(2018) deal with a deep learning model that predicts optimal tactical solutions to the stochastic load planning problem of intermodal containers on double-stack trains. Besides the benefits regarding speed and accuracy of the obtained solution, the authors point out the low requirements in terms of computing power and needed data: they show that, unlike optimization models, the end-to-end learning method succeeds in solving the problem under imperfect information. Abbasi et al. (2020) apply this method to the stochastic optimization problem of blood transshipment in a network of hospitals, aimed at minimizing overall costs. Since commercial solvers and expertise in the optimization field are often not available in these contexts, the authors propose to run the optimization only for a limited number of days, with the support of an external partner, aiming to gather a data sample for the training of a supervised ML model that can be later applied to support daily transshipment decisions. They consider different ML techniques (i.e., k-nearest neighbor, classification and regression tree, random forest, multilayer perceptron artificial neural network) and they apply a heuristic algorithm to turn the solution obtained through the ML model into a feasible one, concluding that the end-to-end learning method always allows to achieve total costs which are close to the optimal ones, also compared to the empirical approach currently used by the hospitals.

3 Combined optimization-ML approach for the ALFP

In this section, we describe an approach that exploits ML techniques to learn the optimal ALFP solution and define simple rules for the selection of part feeding policies in an assembly system.

We consider the problem of selecting feeding policies aiming at the optimization of overall cost or time performance and we refer to the general setting described by Schmid and Limère (2019), where an assembly line, made of several stations, is fed with multiple parts. We assume that information about the assembled products is known in advance and that the design of the assembly line, including line balancing, has already been carried out. A part is intended as any component or sub-assembly required for the assembly of finished products. Each part must be supplied to

a station according to one of five possible feeding policies: line stocking, boxed-supply, sequencing, stationary kitting, or travelling kitting. If the same part is used in more than one station, it can be supplied to different stations according to different policies.

As shown in Fig. 2, our approach is made of four steps. In the first two steps, an optimization model is applied to a sample of parts and its output is used to train a supervised ML model, according to the end-to-end learning method. In the following steps, the ML model is used to select the optimal feeding policy for new parts.

In **Step 1**, an optimization model is run to assign a sample of parts to feeding policies. As discussed in Sect. 2, several models have been presented in previous literature and they are all suitable to be used within this approach. Without loss of generality, the optimization model can be expressed as the minimization of an objective function $y=f(X)$, subject to a number of constraints. The objective function corresponds to the cost or time performance measure selected for the specific application and its value depends on the tensor $X=x_{111}, \dots, x_{NMS}$, where N is the number of parts included in the optimization, M is the number of feeding policies, S is the number of stations, and x_{ijs} is a Boolean variable equal to 1 if part i is supplied to station s according to policy j and equal to 0 otherwise. In general, the constraints include: resource constraints (e.g., restrictions on the storage space available at the border of the line or on the capacity of transport equipment), time constraints (i.e., restrictions on the time available for part feeding activities), ergonomics constraints (i.e., limitations on the ergonomics stress for a single operation or during a shift), and consistency constraints (e.g., definition of Boolean or integer variables). The required input data are all the parameters needed to run the optimization model. The output is the optimal allocation of parts to feeding policies, X^* .

In **Step 2**, the optimization output is used to train the supervised ML model, thus identifying the relevant factors and the classification rules for feeding policies selection. In the input dataset, each part included in Step 1 represents an instance, described by a target variable and by a set of explanatory attributes. The target variable is the optimal feeding policy found in Step 1. The explanatory attributes correspond to part and assembly system characteristics: while it is possible to include in the dataset all the

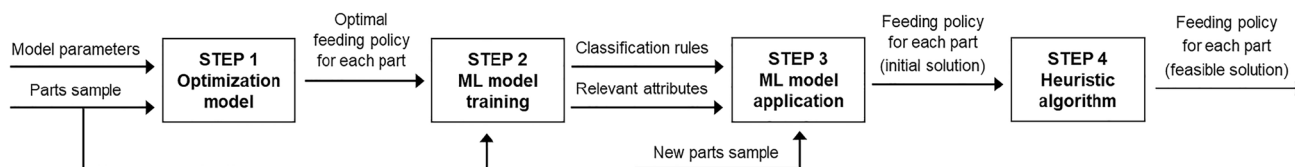


Fig. 2 Combined optimization-ML approach for the ALFP

available data (e.g., all the parameters used in Step 1), we suggest selecting only significant ones based on the knowledge of the specific problem, thus avoiding the risk of low interpretability and accuracy due to an overly complex ML model (Fischetti and Fraccaro 2019). We also suggest performing a transformation of some explanatory attributes (e.g., Fischetti and Fraccaro 2019; Vercellis 2009). In some cases, this transformation could consist only in standardizing the values of certain attributes or changing their unit of measure; for instance, in contexts where the volumes of parts and unit loads vary in very broad ranges, it is useful to measure the parts volume in terms of equivalent number of unit loads. In other cases, new attributes can be created, resulting from the combination of the original ones, to provide the ML model with more relevant information. For instance, this option is useful to let the ML model better learn the optimization model constraints; an example is shown in Sect. 4.

In the training phase, the ML model learns the rules that relate each part to a feeding policy based on the explanatory attributes. Given the presence of both numerical (e.g., part weight) and categorical (e.g., type of unit load) attributes, a classification model is needed for our purpose rather than a regression one. Among the different types of classification models, decision trees (DTs) are particularly suitable for our aim. In fact, unlike other models that act as a “black box”, DTs ensure a good interpretability of the classification rules, which are organized in the form of a tree structure and can be easily understood and applied also by people who are not experts in the ML field (Kotsiantis 2013). Such classification rules are made of a set of relations of the form $y > b$, $y < b$, or $y = b$, where y is a generic explanatory attribute and b is a threshold value selected for that attribute.

The DT consists of a tree structure whose leaf nodes represent the feeding policies and whose branches represent the classification rules that can be followed, proceeding in a top-down fashion, to assign each part to a feeding policy. During the DT training phase, the nodes are progressively split, starting from a root node including all the instances. As regards splitting rules, we suggest building a univariate tree in which the attribute used for each split is chosen through a statistical test of independence, so as to reduce bias in the attribute selection in the presence of both categorical and numerical attributes (Loh and Shih 1997). Instead, the choice of the partitioning criterion should aim at minimizing the impurity of child nodes (Kotsiantis et al. 2007).

The training of the ML model is an iterative process, during which alternative choices, concerning attributes selection, attributes transformation, and splitting criteria setting, should be evaluated in terms of accuracy. In the training phase, the accuracy can be assessed through the cross validation error, thus reducing the effect of the test set selection on the model performance evaluation (Vercellis 2009).

After building a full-grown tree, pruning criteria are applied to improve accuracy and interpretability by merging leaves on the same branch. The chosen pruning level can be set as the one that minimizes the cross validation error or as the highest level for which the cross validation error does not exceed the minimum error plus a number of standard deviations. This last option is useful to improve the tree interpretability and prevent it from overfitting the training set, but a low number of standard deviations (e.g., one or two) is suggested to keep a good accuracy. As a result of pruning, only the attributes that contribute the most to explain the ALFP solution are left in the DT. At the end of the pruning procedure, each leaf node is labelled with a part feeding policy, chosen based on the majority of the training instances in the node.

Once developed, the DT can be applied to estimate the optimal ALFP solution through two successive steps (Fig. 2). In **Step 3**, the classification rules, learnt in Step 2, are applied to assign new parts to feeding policies. Only data concerning the attributes found to be relevant after the DT training must be collected to get this solution. The output of Step 3 is an initial ALFP solution, given in input to Step 4.

In **Step 4**, a check is performed to verify whether the initial solution satisfies the constraints included in the optimization model. The feasibility of the ML model solution is one of the challenges in the research on combined optimization-ML approaches (Bengio et al. 2020) and, so far, only heuristic algorithms have been used to turn the solution obtained through the ML model into a feasible one (Abbasi et al. 2020). Therefore, in Step 4 a heuristic algorithm is employed, consisting of an iterative procedure that prescribes to progressively change the assignment of parts to feeding policies until the constraints are satisfied. At each iteration, the assignment of one part is changed by replacing the initial policy with a new one that contributes to satisfy the constraint. In order to make this procedure more efficient and to avoid excessive deviations from the initial solution, parts can be sorted according to the value of one or more parameters. For instance, in case the capacity constraint of tigger trains is not satisfied, parts are progressively moved from the boxed-supply policy to the line stocking one (provided that the latter does not require tigger trains for parts transportation). In this case, parts can be sorted by decreasing share of tigger train capacity they take up.

The model development steps (Steps 1 and 2) are required to be performed only once and considering a sample of parts. Such sample can correspond either to the whole assembly system or to a subsystem, representative of the whole system in terms of possible values of the explanatory attributes. For instance, in case an assembly plant is composed of several assembly lines, similar in terms of operating conditions (e.g., layout, automation level) and product models, only one or few lines could be considered for the model

development. The last two steps, instead, can be applied to solve the ALFP in the remaining subsystems, if any, as well as in the future, every time new product models or part variants are introduced.

4 Application of the proposed approach

In this section, we show an application of the approach introduced in Sect. 3. In order to make our results comparable with previous literature, we consider the setting and optimization model provided in Limère et al. (2015).

4.1 Problem description and parameters

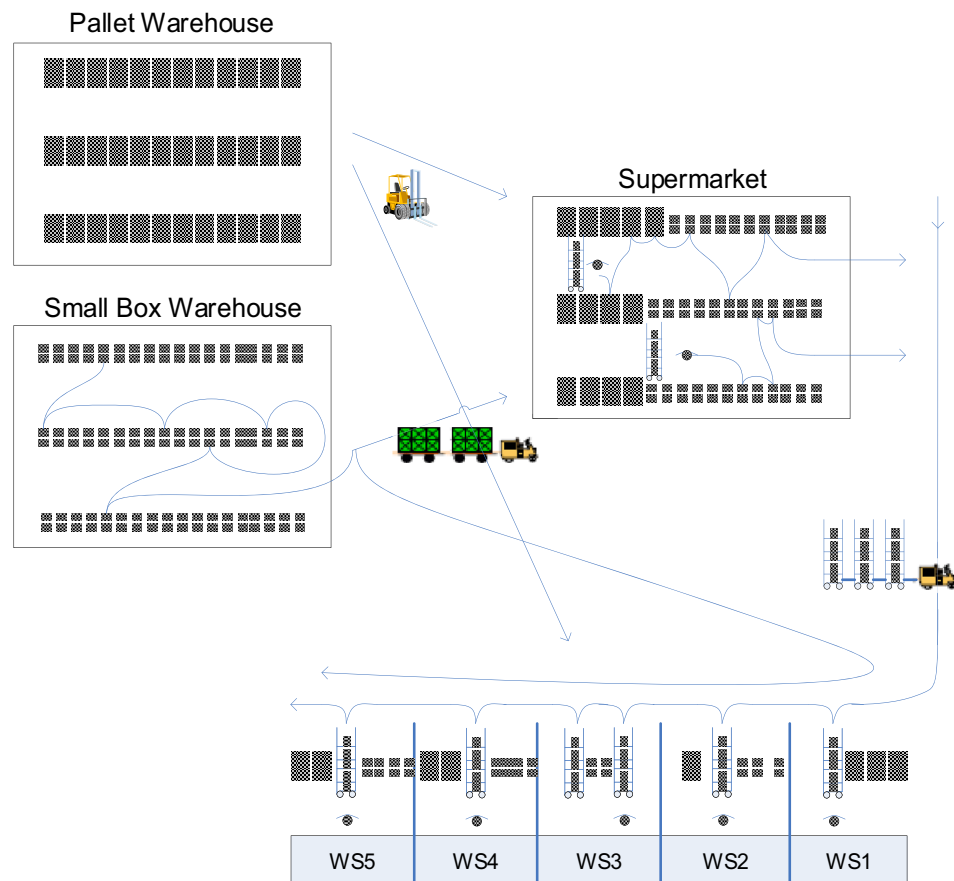
The application concerns a European truck manufacturing company which is selecting a feeding policy for each part family in a plant producing medium duty trucks. A part family is made of two or more variants of the same part, identical in terms of physical characteristics and BOM coefficients, thus differing only for the demand and the number of finished product models in which the specific variant is used. The plant layout, schematically depicted in Fig. 3, consists of two central warehouses (a high bay warehouse for pallets

and a small bay warehouse for smaller boxes), a supermarket area, and a number of assembly stations.

Two part feeding policies are used: line stocking and stationary kitting (called *kitting* in the following). If line stocking is adopted, the homogeneous unit loads (ULs) received from suppliers or upstream production stages (called *original ULs* in the following) are delivered to the assembly stations directly from the central warehouses. If the original UL is a pallet or a big container, the internal transportation is performed by means of forklift trucks and the replenishment is controlled through a reorder-point system. If the original UL is a box or a small container, trolley trains are employed for the internal milk run transportation and the replenishment is controlled through a two-bin inventory system. When reaching the border of the assembly stations, original ULs are stored on the shop floor until they have been completely used up. While pallets cannot be stacked, four boxes can be stored in the same column at the border of a station.

In case kitting is chosen, stationary kits are prepared in the supermarket area before being transported to the border of the assembly stations. The supermarket is a warehouse storing both pallets and boxes, replenished from the central warehouses. Pickers walk the supermarket aisles and prepare kits according to a picker-to-part configuration. A kit is made of a set of parts used at an assembly station for the

Fig. 3 Plant layout



assembly of one truck. A kit container consists of a rack with multiple levels: thus, a fixed number of kits are included in each kit container and they are prepared in batches inside the supermarket. Kits are transported to the assembly stations by means of tugger trains, according to milk run tours. Since one kit is used per takt time, kit containers are replenished at constant time intervals.

For this research, we consider the dataset used by Limère et al. (2015), made of 8,905 parts and corresponding to five mixed-model assembly lines similar to each other in terms of layout and product models they assemble. All the general parameters are reported in Table 3.

4.2 ML model training

In **Step 1**, we run the optimization model by Limère et al. (2015) to obtain the optimal feeding policies for a sample of parts. Given that the five assembly lines are similar to each other, we apply the optimization model only to a representative subsystem made of two lines (called *Line 1* and *Line 2* in the following), corresponding to 3,641 parts. The mixed integer linear programming model aims at assigning each part family to either line stocking or kitting in order to minimize the overall costs, computed as the sum of four cost items: supermarket replenishment, internal transportation, kit preparation, and picking at the

line. Storage space constraints at the border of each station are included in the model, as well as consistency constraints (i.e., definition of Boolean variables, constraints to linearize the problem and to ensure that all variants in a family are assigned to the same policy, kit volume and weight capacity restrictions).

In **Step 2**, we first prepare the input dataset for training the DT. This dataset consists of the list of the part families considered in Step 1 reporting, for each family, the target variable (i.e., the optimal feeding policy) and the explanatory attributes. The complete list of attributes is shown in Table 4. The majority of them (10 out of 12) are considered in previous literature (Table 2), while the remaining 2 result from an attribute transformation: the storage space occupied by the part family at the border of the station if the family is assigned to line stocking (*Space_fam* in the following), obtained as the combination of the original UL length and the part family cardinality; the storage space occupied at the border of the station if all the families supplied to the station are assigned to line stocking (*Space_stat* in the following), obtained as the combination of the original UL length and the cardinality of the station. Such new attributes are included to improve the DT capability to learn the storage space constraint: the higher their values, the lower the share of part families that can be assigned to line stocking in order to satisfy the constraint.

Table 3 General problem parameters (Adapted from Limère et al. 2015)

Parameter	Value
Average walking speed of an operator [m/s]	1
Average distance for the supermarket operator to pick parts from an original UL [m]	1.5
Average time to search for the required part from an original UL [s]	1.08
Fixed component of the kit preparation time [s]	10
Batch size for preparing kits	5
Distance of a milk run tour for boxes [m]	1,640
Capacity of the milk run tours for boxes (number of boxes per tour)	60
Expected capacity utilization of the milk run tours for boxes	50%
Distance of the milk run tour for kits [m]	1,640
Capacity of the milk run tours for kits (number of kits per tour)	70
Expected capacity utilization of the milk run tours for kits	80%
Average velocity of the material handling equipment for milk run tours [m/s]	0.67
Average velocity of the forklift trucks [m/s]	0.8
Fixed cost for the replenishment of one box in the supermarket [€/box]	0.2
Fixed cost for the replenishment of one pallet in the supermarket [€/pallet]	1.2
Weight constraint on one kit (maximum weight per kit) [kg]	50
Average distance for the line operator to pick from a kit [m]	1.5
Length of a pallet along the border of an assembly station [m]	1
Length of a stack of boxes box along the border of an assembly station [m]	0.8
Length of a kit container along the border of an assembly station [m]	0.8
Storage space available at the border of one assembly station [m]	8
Hourly cost of labor [€/h]	30

Table 4 Explanatory and relevant attributes

Explanatory attributes used for the training phase	Relevant attributes included in the DT
Unit weight	
Unit volume (maximum number of units of the part a kit can hold)	X
Maximum number of units per pick	X
BOM coefficient (number of parts per finished product)	X
Type of original UL (pallet or box)	
Number of part units per original UL	X
Cardinality of the part family (number of part variants belonging to the family)	X
Yearly demand of the part family	
Percentage of product models for which the part family is required	
Cardinality of the station to which the part family is supplied (overall number of parts supplied to the station)	X
Storage space occupied by the part family at the border of the station if the part family is assigned to line stocking (<i>Space_fam</i>)	X
Storage space occupied at the border of the station if all the part families supplied to the station are assigned to line stocking (<i>Space_stat</i>)	X

We train the DT through the software MATLAB. We measure the accuracy according to a k -fold cross validation method: after trying different values of the parameter k , we set it to 500, which is a number of folds high enough to achieve a stable DT, not affected by the choice of training and test sets. As for the splitting criteria, we choose the explanatory attribute used for each split according to a statistical test of independence. We run chi-square tests of independence both between each attribute and the target variable (curvature test) and between each pair of attributes and the target variable (interaction test). Interaction tests are useful since they allow to account for relevant correlations between the attributes created through transformation and the attributes combined to create them. In case the lowest significant p-value results from a curvature test, we use the related attribute for the split; otherwise, looking at the pair of attributes in the interaction test with the lowest p-value, we choose the one that maximizes the impurity gain after the split; finally, in case all the p-values are not significant (i.e., higher than 0.05), the node is not further split. With regard to the partitioning criterion, we compare three alternatives (Gini index, entropy index, twoing rule) and we select the Gini index since it leads to the lowest cross validation error in the final tree. After building a full-grown tree, we assess all possible pruning levels and select the highest one for which the cross validation error ($k = 500$) does not exceed the minimum error plus one standard deviation. The final ML model is reported in Fig. 4, that shows the classification rules and the relevant attributes.

The first splitting rule prescribes to adopt line stocking if the overall space occupied by all the parts used at a station is lower than the available one; otherwise, the second rule leads to different classifications depending on the volume

of each part. The following rules involve 6 more attributes, adding up to a total of 8 relevant attributes out of the 12 used for the training (Table 4). The two transformed attributes, *Space_stat* and *Space_fam*, are both included, confirming that attribute transformation can help explain the optimal ALFP solution.

4.3 ML model application

In **Step 3**, we use the developed DT to select feeding policies for the parts used by the remaining three assembly lines (called *Line 3*, *Line 4*, and *Line 5* in the following and corresponding to 5,264 parts), considering the 8 relevant attributes found in Step 2. At the end of Step 3, we obtain an initial solution where 53% of the part families are assigned to line stocking and the remaining ones to kitting.

In **Step 4**, we check the feasibility of the initial solution. The workstations not meeting the storage space constraint are 3.56% of the total, indicating that the DT has successfully learnt most of the constraints. Although most of these infeasibilities could be solved with space borrowing (Schmid et al. 2021), we apply the heuristic algorithm reported in the Appendix. As a result, we get a feasible solution, close to the previous one, where 52% of the part families are assigned to line stocking and 48% to kitting.

5 Performance assessment and discussion

In this section, we explore the trade-off between efficiency and quality of the ALFP solution. In Sect. 5.1, we compare our approach with the optimization model presented in Limère et al. (2015). Then, we show the effect of a reduction

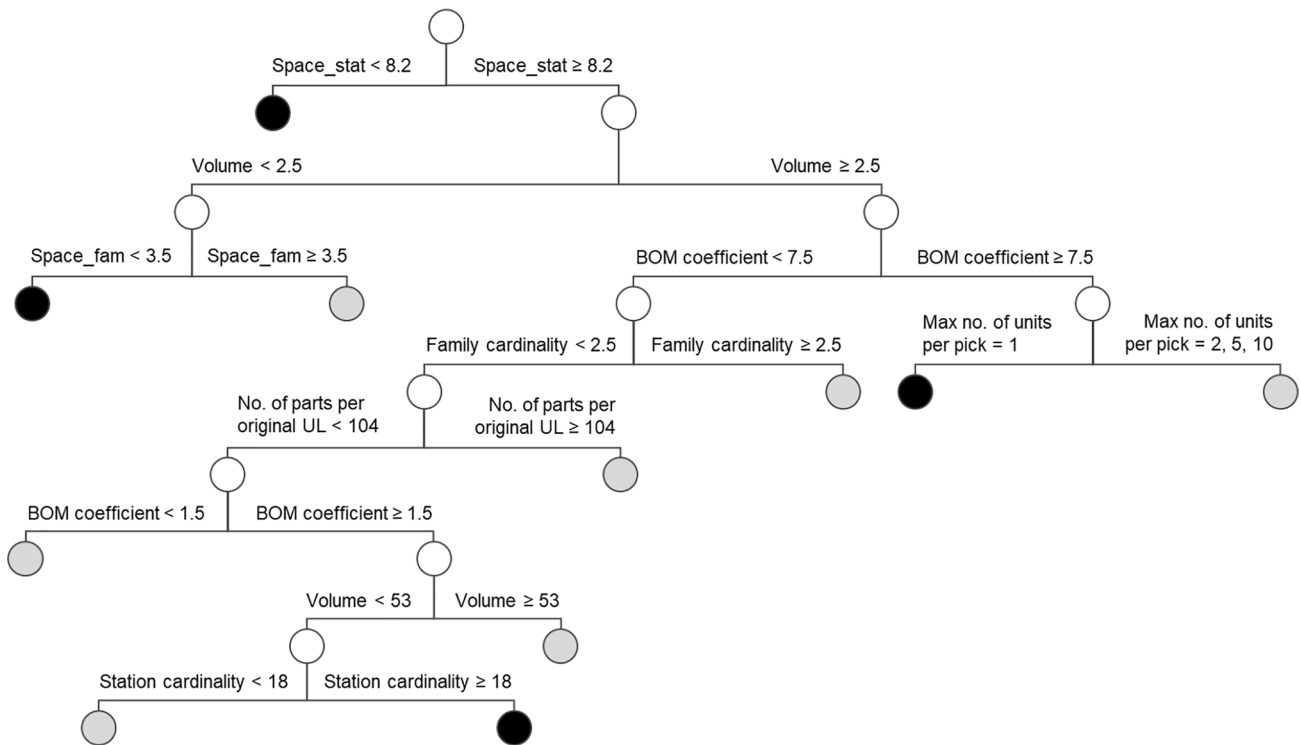


Fig. 4 Final DT

in the data collection effort. We study how the solution changes when decreasing the number of relevant attributes used to solve the ALFP (Sect. 5.2), the size of the subsystem used to train the ML model (Sects. 5.3), or both (Sect. 5.4). We perform these analyses considering the parts supplied to Lines 3, 4, and 5, not used for the ML model training. Based on the results of these analyses, we discuss the effectiveness of the proposed approach and how it compares with the methods presented in previous literature (Sect. 5.5).

5.1 Comparison with the optimal solution

We compare the ALFP solution resulting from our approach with the optimal one, both in terms of accuracy and costs. We measure accuracy as the percentage of part families assigned to the same policy by the ML and optimization

models. As regards costs, we consider the percentage difference between the yearly costs corresponding to the ALFP solution obtained through the ML model and the optimal yearly costs. To compute these costs, we use the objective function described in Sect. 4.2.

First, we consider the initial solution (i.e., Step 3 output): Table 5 reports the results for each assembly line. Overall, the optimization and ML models give the same outcome in terms of feeding policy assignments for 83.1% of the part families. The 16.9% classification error is split between families wrongly assigned to kitting by the DT (9.2%), which are expected to generate extra costs mainly due to picking activities in the supermarket, and families wrongly assigned to line stocking (7.7%), which tend to reduce overall costs but contribute to cause space constraints infeasibilities. The difference between ML and optimization models is even

Table 5 Comparison between ML (initial solution) and optimization

Assembly line	Accuracy	Yearly costs (Optimization model)	Yearly costs (ML model, Step 3)	Cost percentage difference
Line 3	84.0%	371,866 €	398,004 €	7.0%
Line 4	82.6%	397,297 €	426,461 €	7.3%
Line 5	82.4%	455,856 €	475,564 €	4.3%
Total	83.1%	1,225,019 €	1,300,029 €	6.1%

Table 6 Comparison between ML (feasible solution) and optimization

Assembly line	Accuracy	Yearly costs (Optimization model)	Yearly costs (ML model, Step 4)	Cost percentage difference
Line 3	84.4%	371,866 €	398,762 €	7.2%
Line 4	82.5%	397,297 €	430,019 €	8.2%
Line 5	83.8%	455,856 €	486,897 €	6.8%
<i>Total</i>	<i>83.6%</i>	<i>1,225,019 €</i>	<i>1,315,678 €</i>	<i>7.4%</i>

lower looking at costs: the average costs obtained through the DT are 6.1% higher than the optimal ones.

Table 6 reports the results concerning the feasible solution. It shows that the accuracy slightly improves thanks to the application of the heuristic algorithm to the stations not satisfying the storage space constraint, where some parts, initially wrongly assigned to line stocking, are assigned to kitting. An opposing effect is registered with regard to costs, which are 7.4% higher than the optimal ones. This result can be considered acceptable also if compared with previous literature, where the only available benchmark shows a 20.5% cost increase if a DT is used to estimate the optimal solution (Abbasi et al. 2020).

5.2 Effect of the number of relevant attributes

With refer to the feasible solution, we study how the number of relevant attributes used to solve the ALFP affects the solution quality when the combined optimization-ML approach is adopted. Starting from the DT depicted in Fig. 4, we progressively increase the pruning level. At each step, we remove the attributes placed closer to the tree leaves, which are the ones showing the lowest correlation with the optimal assignment of parts to feeding policies, up to the point when only the two most relevant attributes are left.

As reported in Table 7, with each additional pruning level one or more attributes are removed from the DT, meaning that the related data does not have to be collected for the 5,264 parts used in Lines 3, 4, and 5. Table 7 also shows that reducing the amount of data which has to be gathered to apply the ML model leads to higher overall costs. However, costs are quite robust to the decrease in the data collection effort. For instance, when using 5 attributes instead of 8, the amount of data needed to solve the ALFP is reduced by 37.5% while costs increase by 1.8%. Even in the case when only 2 relevant attributes are considered, costs are only 5.2% higher than in the case with no additional pruning.

5.3 Effect of the training dataset size

With refer to the feasible solution, we study how the model performance changes when reducing the training dataset size. We compare the training dataset considered so far, made of the part families used by Lines 1 and 2, with two alternative ones. The first one includes the part families used by one assembly line only (Line 1). The second one includes half of the part families used by Line 1, randomly chosen.

As shown in Table 8, when training the ML model with data concerning Line 1 only (thus, almost halving the data collection effort compared to the first setting),

Table 7 Model performance varying the number of relevant attributes

Number of relevant attributes	Relevant attributes	Additional pruning levels	Accuracy	Total yearly costs	Amount of data (ML model application)*
8	Unit volume, Maximum no. of units per pick, BOM coefficient, Number of part units per original UL, Family cardinality, Station cardinality, Space_fam, Space_stat	(no additional pruning)	83.1%	1,315,678 €	42,112
5	Unit volume, Maximum no. of units per pick, BOM coefficient, Space_fam, Space_stat	1	82.4%	1,338,787 €	26,320
4	Unit volume, BOM coefficient, Space_fam, Space_stat	2	80.4%	1,367,528 €	21,056
3	Unit volume, Space_fam, Space_stat	3	80.9%	1,392,591 €	15,792
2	Unit volume, Space_stat	4	81.1%	1,384,697 €	10,528

*Product between the number of parts and the number of relevant attributes

Table 8 Model performance varying the training dataset size

Training dataset size	Amount of data (ML model training)*	Accuracy	Total yearly costs
Part families used by Lines 1, 2	40,051	83.1%	1,315,678 €
Part families used by Line 1	19,965	82.4%	1,326,807 €
Half of the part families used by Line 1	10,725	77.1%	1,401,631 €

*Product between the number of parts and the number of explanatory attributes

performance remains almost unchanged in terms of both accuracy and costs. Instead, in case a subset of the parts used by Line 1 is used to train the ML model, we observe a decrease in the accuracy and a significant increase in overall costs, which become 14.4% higher than the optimal ones.

These results confirm that the performance of the ML model in solving the ALFP remains almost unchanged as long as the subsystem used to develop the model is representative of the whole system in terms of possible values of the explanatory attributes; in this case, being the five lines very similar in terms of layout and assembled models, it is sufficient to use only one of them as representative subsystem. On the contrary, the performance worsens if this condition is no longer satisfied.

5.4 Joint effect of the number of relevant attributes and training dataset size

With refer to the feasible solution, we study how the model performance changes if the number of relevant attributes and the training dataset size are jointly reduced. We consider the same training dataset sizes investigated in Sect. 5.3 and, for each one, we compare the case in which all relevant attributes resulting from the ML model training are used to solve the ALFP, with the case in which only the two most relevant attributes are used.

Table 9 reports the accuracy and total yearly costs for each of the considered combinations of training dataset size and number of relevant attributes.

Obviously, the case with maximum training dataset size and number of relevant attributes leads to the best performance. Then, whatever the considered training dataset size, a decrease in the number of relevant attributes allows to solve the ALFP with a significantly lower data collection effort, without heavily worsening overall performance. The variation in the accuracy is below 2% when considering only two relevant attributes. The increase in overall costs is also explained by the fact that, in this particular setting, the classification error is mostly related to parts wrongly assigned by the ML model to kitting policy, which is more costly than line stocking.

This analysis also confirms that the training dataset size can be reduced without significantly affecting performance, as long as the subsystem corresponding to the training dataset is representative of the whole system in terms of possible values of the explanatory attributes. As shown in Table 9, when considering two attributes, the same performance is achieved by training the ML model with either the data corresponding to Lines 1 and 2 or the data corresponding to just Line 1: this happens because the ML model training leads to the identification of the same two most relevant attributes in both cases. Instead, the two most relevant attributes selected during the training change when only half of

Table 9 Model performance varying the number of relevant attributes and the training dataset size

Training dataset size	All relevant attributes	Two relevant attributes
Part families used by Lines 1, 2	Accuracy: 83.1% Total yearly costs: 1,315,678 € Amount of data (training): 40,051 Amount of data (application): 42,112	Accuracy: 81.1% Total yearly costs: 1,384,697 € Amount of data (training): 40,051 Amount of data (application): 10,528
Part families used by Line 1	Accuracy: 82.4% Total yearly costs: 1,326,807 € Amount of data (training): 19,965 Amount of data (application): 42,112	Accuracy: 81.1% Total yearly costs: 1,384,697 € Amount of data (training): 19,965 Amount of data (application): 10,528
Half of the part families used by Line 1	Accuracy: 77.1% Total yearly costs: 1,401,631 € Amount of data (training): 10,725 Amount of data (application): 42,112	Accuracy: 79.6% Total yearly costs: 1,449,019 € Amount of data (training): 10,725 Amount of data (application): 10,528

Fig. 5 Algorithm

Algorithm	
1:	for each station where space constraints are not satisfied do
2:	for each part family do
3:	compute the number of kits K_s necessary for the part family, if it was assigned to kitting
4:	end for
5:	while the space constraint is not satisfied do
6:	sort the part families in decreasing order of cardinality
7:	for each part family assigned to line stocking, whose original UL is a pallet do
8:	if the overall space occupied at the border of the station decreases by assigning the part family to kitting then
9:	assign the part family to kitting
10:	end if
11:	end for
12:	sort the part families in order of increasing value of $B = \text{part family cardinality} / K_s$
13:	for each part family assigned to kitting do
14:	if the overall space occupied at the border of the station decreases by assigning the part family to line stocking then
15:	assign the part family to line stocking
16:	end if
17:	end for
18:	end while
19:	end for

the parts provided to Line 1 are used to train the model, confirming that this smaller dataset does not enable the ML model to successfully learn the optimal assignment of parts to feeding policies.

5.5 Discussion

As shown in the literature review (Sect. 2.1), the existing methods to solve the ALFP imply a trade-off between solution quality and efficiency. Inductive approaches allow finding the best combination of policies looking at their effect on system performance, but their application might be difficult as they require a large amount of data and specific skills and software. Conversely, deductive approaches, based on parts categorization, are easy to adopt as they entail a low data collection effort and they are intuitive and ‘ready-to-use’ when new product models or part variants are introduced. However, their main drawback lies in the fact that they do not ensure to achieve a performance close to the optimal one.

The application reported in Sect. 4 confirms that the approach proposed in this paper represents a new effective way to select part feeding policies in an assembly system, improving the trade-off emerged in previous literature. On the one hand, similarly to the methods based on parts categorization, in the proposed approach parts are assigned to feeding policies according to simple and intuitive rules, based on a few attributes of the parts and the assembly system. Therefore, shop floor managers can quickly update the ALFP solution every time new parts or product models are

introduced, without the support from experts in the optimization or ML domains. On the other hand, the use of ML ensures that the attributes on which the solution is based are the relevant ones, explaining the optimal assignment of parts to feeding policies. As a consequence, the achieved solution is close to the optimal one in terms of both accuracy and costs (16.9% and 7.4% difference, respectively, in the case considered in this paper).

Moreover, the proposed approach can be easily adapted to be applied to industrial cases across several industries where manual assembly operations are carried out on a line, such as consumer electronics, household appliances, healthcare equipment, industrial and agricultural machinery. Unlike the approaches presented in literature, according to which the ALFP solution is based on a pre-defined set of performance measures to be optimized (as shown in Table 1 in the Literature Review Section) or on a pre-defined set of factors (as shown in Table 2 in the Literature Review Section), the one proposed in this paper is extremely flexible because it works with any optimization model and considering any factors, chosen depending on their relevance and availability in the considered industrial case.

Finally, as regards the data collection effort, the proposed approach can be placed in between the two classes of methods presented in literature. In fact, the optimization model parameters must be gathered, when training the ML model, only for a subsystem. For the remaining portion of the assembly system, as well as when new parts are introduced, the only data needed to select feeding policies are the

ones concerning a few relevant attributes, whose number can be set according to data availability in the specific context (see Sects. 5.2 and 5.4). Therefore, in case of scarce data availability a key issue lies in the choice of the subsystem used for the ML model training, which should be performed in close collaboration with the plant managers. In fact, as shown in Sects. 5.3 and 5.4, a poorly selected subsystem could turn out not to be representative of the whole system in terms of possible values of the explanatory attributes, thus hindering the ML model performance.

6 Conclusions

6.1 Contribution

In this paper, we explore the application of ML to part feeding policies selection in assembly systems. We present a new approach to solve the ALFP, based on the end-to-end learning method, according to which parts are assigned to feeding policies through a supervised ML model trained using the output of an optimization model. We show how this approach can be concretely applied and how it performs compared to optimization approaches. Finally, we discuss how performance changes when varying the quantity of data used in the ML model training and application.

From an academic point of view, this paper enriches the previous literature both on part feeding policies selection and on the application of ML techniques in production systems. Indeed, we introduce a new approach to solve the ALFP that is general in the sense that it can be implemented in any assembly system, considering any number of feeding policies. We show how this approach enables to improve the trade-off between the effort required to solve the ALFP and the quality of the solution, by pairing the advantages of the methods currently available in literature. Moreover, we broaden the application field of ML techniques in production systems, by both considering the ALFP and introducing a new type of approach, based on the end-to-end learning method, applied so far only in different research areas and by very few contributions. At last, we contribute to advance the knowledge on combined optimization-ML approaches since we formalize, for the first time in literature, the steps required for the application of such approaches and address issues such as the feasibility of the ML model solution and its comparison with the optimal solution, which are still understudied.

From a managerial point of view, the approach proposed in this paper represents an effective method to select feeding policies in contexts where data availability is critical or data collection is very time-consuming. As shown through the application to the industrial case, the ML model

performance is robust to the number of attributes used to select feeding policies. Moreover, the size of the subsystem used for the ML model training can be reduced as long as it remains representative of the overall assembly system. Besides being robust to the amount of available data, the proposed approach allows managers to estimate the optimal ALFP solution in short time and with low effort, using intuitive classification rules which are similar to the ones already applied in most assembly systems. Most of the effort, in terms of time, skills, and computing power, is concentrated in the ML model development steps, which must be performed only once and possibly with the support of consultants or research institutes, as suggested by Abbasi et al. (2020) with refer to a different end-to-end learning approach.

6.2 Limitations and future research

Being a first attempt to explore the adoption of ML to solve the ALFP, this study presents some limitations, which generate opportunities for further research. First of all, the developed approach is applied to only one industrial case. Future research should demonstrate performance in different contexts. For instance, it could account for assembly systems where more than two feeding policies are adopted, thus considering a different optimization model and a wider set of factors. Moreover, while this study has shown that the sample of parts used to train the ML model has a relevant impact on the model performance, the presented numerical analysis about the size of the representative training dataset is specific to the case under investigation. Therefore, future research should carry out a more in-depth investigation of this matter, aiming to define general criteria for sample selection. Next, we evaluate a single ML technique, the DT, chosen mainly due to its good interpretability. Future research could investigate the application of different techniques characterized by a higher accuracy, such as random forests and neural networks, paying attention to the trade-off between the increased accuracy and the lower interpretability (Kotsiantis 2013). In addition, while this study carries out a comparison between the proposed approach and an optimization-based one, a future development could be the comparison and integration of the proposed approach with those currently used by practitioners. A further limitation concerns the fact that the approach has been developed considering a setting in which the preliminary design of the assembly system, including its balancing, has already been carried out. An interesting direction for future research is the extension of this approach to support the integrated assembly line balancing and feeding problems (Sternatz 2015; Schmid and Limère 2019). Finally, since ML is often considered as one of the main enablers for the evolution of a traditional manufacturing system into a 4.0 system (Culot et al. 2020; Bertolini et al. 2021), this work is connected to the research on Industry 4.0, which aims at improving the efficiency and

flexibility of production processes thanks to the collection, sharing, and analysis of data (Ghobakhloo 2018; Garay-Rondero et al. 2019; Oztemel and Gursev 2020). However, this study does not explore the interplay between ML and other Industry 4.0 enabling technologies (Culot et al. 2020). In future research, the proposed approach could be integrated with technologies supporting real-time data collection and information sharing. For instance, it could be adapted into a more operational decision tool within a hyper-connected system, where feeding policies could be dynamically assigned to parts based on real-time data fed to the ML model, concerning the shop floor status (e.g., machines failures or disruptions which make one of the feeding policies not viable for a certain period of time) and the actual demand level.

Appendix

Heuristic algorithm to solve storage space constraint infeasibilities.

Authors contributions All the authors were involved in the research conception and design, in data analysis, as well as in the analysis and discussion of results. All the authors wrote a part of the manuscript and approved the submitted manuscript.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement. No funding was received for conducting this study.

Availability of data and material The dataset used for this research is available upon request.

Code availability The MATLAB code used for this research is available upon request.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abbasi B, Babaei T, Hosseini Z, Smith-Miles K, Dehghani M (2020) Predicting solutions of large-scale optimization problems via machine learning: A case study in blood supply chain management. *Comput Oper Res* 119. <https://doi.org/10.1016/j.cor.2020.104941>
- Baller R, Hage S, Fontaine P, Spinler S (2020) The assembly line feeding problem: An extended formulation with multiple line feeding policies and a case study. *Int J Prod Econ* 222. <https://doi.org/10.1016/j.ijpe.2019.09.010>
- Battini D, Faccio M, Persona A, Sgarbossa F (2009) Design of the optimal feeding policy in an assembly system. *Int J Prod Econ* 121(1):233–254. <https://doi.org/10.1016/j.ijpe.2009.05.016>
- Bello I, Pham H, Le QV, Norouzi M, Bengio S (2016) Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940.
- Bengio Y, Lodi A, Prouvost A (2020) Machine learning for combinatorial optimization: a methodological tour d'horizon. *Eur J Oper Res*, in Press. <https://doi.org/10.1016/j.ejor.2020.07.063>
- Bertolini M, Mezzogori D, Neroni M, Zammori F (2021) Machine Learning for industrial applications: a comprehensive literature review. *Expert Syst Appl* 175:114820. <https://doi.org/10.1016/j.eswa.2021.114820>
- Bozer YA, McGinnis LF (1992) Kitting versus line stocking: a conceptual framework and a descriptive model. *Int J Prod Econ* 28(1):1–19. [https://doi.org/10.1016/0925-5273\(92\)90109-K](https://doi.org/10.1016/0925-5273(92)90109-K)
- Caputo AC, Pelagagge PM (2011) A methodology for selecting assembly systems feeding policy. *Ind Manag Data Syst* 111(1):84–112. <https://doi.org/10.1108/02635571111099749>
- Caputo AC, Pelagagge PM, Salini P (2015) A decision model for selecting parts feeding policies in assembly lines. *Ind Manag Data Syst* 115(6):974–1003. <https://doi.org/10.1108/IMDS-02-2015-0054>
- Caputo AC, Pelagagge PM, Salini P (2018) Selection of assembly lines feeding policies based on parts features and scenario conditions. *Int J Prod Res* 56(3):1208–1232. <https://doi.org/10.1080/00207543.2017.1407882>
- Culot G, Nassimbeni G, Orzes G et al (2020) Behind the definition of Industry 4.0: Analysis and open questions. *Int J Prod Econ* 226:107617. <https://doi.org/10.1016/j.ijpe.2020.107617>
- Emde S, Schneider M (2018) Just-in-time vehicle routing for in-house part feeding to assembly lines. *Transp Sci* 52(3):657–672. <https://doi.org/10.1287/trsc.2018.0824>
- Faccio M (2014) The impact of production mix variations and models varieties on the parts-feeding policy selection in a JIT assembly system. *The International Journal of Advanced Manufacturing Technology* 72(1–4):543–560. <https://doi.org/10.1007/s00170-014-5675-0>
- Faccio M, Gamberi M, Bortolini M, Pilati F (2018) Macro and micro-logistic aspects in defining the parts-feeding policy in mixed-model assembly systems. *Int J Serv Oper Manag* 31(4):433–462. <https://doi.org/10.1504/IJSOM.2018.096166>
- Fischetti M, Fraccaro M (2019) Machine learning meets mathematical optimization to predict the optimal production of offshore wind parks. *Comput Oper Res* 106:289–297. <https://doi.org/10.1016/j.cor.2018.04.006>
- Garay-Rondero CL, Martinez-Flores JL, Smith NR et al (2019) Digital supply chain model in Industry 4.0. *J Manuf Technol Manag* 31(5):887–933. <https://doi.org/10.1108/JMTM-08-2018-0280>
- Ghobakhloo M (2018) The future of manufacturing industry: a strategic roadmap toward Industry 4.0. *J Manuf Technol Manag* 29(6):910–936. <https://doi.org/10.1108/JMTM-02-2018-0057>

- Golz J, Gujjula R, Günther H, Rinderer S, Ziegler M (2012) Part feeding at high-variant mixed-model assembly lines. *Flex Serv Manuf J* 24(2):119–141. <https://doi.org/10.1007/s10696-011-9116-1>
- Gong D, Gen M, Yamazaki G, Xu W (1995) Neural network approach for general assignment problem. *Proceedings of ICNN'95-Int Conf Neural Networks 4*:1861–1866. <https://doi.org/10.1109/ICNN.1995.488905>
- Hanson R, Medbo L (2019) Man-hour efficiency of manual kit preparation in the materials supply to mass-customised assembly. *Int J Prod Res* 57(11):3735–3747. <https://doi.org/10.1080/00207543.2019.1566653>
- Hopfield JJ, Tank DW (1985) “Neural” computation of decisions in optimization problems. *Biol Cybern* 52(3):141–152. <https://doi.org/10.1007/BF00339943>
- Kang Z, Catal C, Tekinerdogan B (2020) Machine learning applications in production lines: A systematic literature review. *Comput Ind Eng* 149:106773. <https://doi.org/10.1016/j.cie.2020.106773>
- Kotsiantis SB, Zaharakis I, Pintelas P (2007) Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering* 160(1):3–24
- Kotsiantis SB (2013) Decision trees: a recent overview. *Artif Intell Rev* 39(4):261–283. <https://doi.org/10.1007/s10462-011-9272-4>
- Larsen E, Lachapelle S, Bengio Y, Frejinger E, Lacoste-Julien S, Lodi A (2018) Predicting Tactical Solutions to Operational Planning Problems under Imperfect Information. *arXiv preprint arXiv:1807.11876*.
- Limère V, Van Landeghem H, Goetschalckx M, Aghezzaf E, McGinnis L (2012) Optimising part feeding in the automotive assembly industry: deciding between kitting and line stocking. *Int J Prod Res* 50(15):4046–4060. <https://doi.org/10.1080/00207543.2011.588625>
- Limère V, Van Landeghem H, Goetschalckx M (2015) A decision model for kitting and line stocking with variable operator walking distances. *Assem Autom* 35(1):47–56. <https://doi.org/10.1108/AA-05-2014-043>
- Loh W, Shih Y (1997) Split selection methods for classification trees. *Stat Sin* 7(4):815–840
- Ohlsson M, Peterson C, Söderberg B (1993) Neural networks for optimization problems with inequality constraints: the knapsack problem. *Neural Comput* 5(2):331–339. <https://doi.org/10.1162/neco.1993.5.2.331>
- Oztemel E, Gursev S (2020) Literature review of Industry 4.0 and related technologies. *J Intell Manuf* 31(1):127–182. <https://doi.org/10.1007/s10845-018-1433-8>
- Sali M, Sahin E (2016) Line feeding optimization for Just in Time assembly lines: An application to the automotive industry. *Int J Prod Econ* 174:54–67. <https://doi.org/10.1016/j.ijpe.2016.01.009>
- Schmid NA, Limère V (2019) A classification of tactical assembly line feeding problems. *Int J Prod Res* 57(24):7586–7609. <https://doi.org/10.1080/00207543.2019.1581957>
- Schmid NA, Limère V, Raa B (2021) Mixed model assembly line feeding with discrete location assignments and variable station space. *Omega* 102:19. <https://doi.org/10.1016/j.omega.2020.102286>
- Smith KA (1999) Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS J Comput* 11(1):15–34. <https://doi.org/10.1287/ijoc.11.1.151>
- Sternatz J (2015) The joint line balancing and material supply problem. *Int J Prod Econ* 159:304–318. <https://doi.org/10.1016/j.ijpe.2014.07.022>
- Usta SK, Oksuz MK, Durmusoglu MB (2017) Design methodology for a hybrid part feeding system in lean-based assembly lines. *Assem Autom* 37(1):84–102. <https://doi.org/10.1108/AA-09-2016-114>
- Vercellis C (2009) Business intelligence: data mining and optimization for decision making. Wiley Online Library. <https://doi.org/10.1002/9780470753866>
- Vinyals O, Fortunato M, Jaitly N (2015) Pointer networks. *Adv Neural Inf Proces Syst*. <http://papers.nips.cc/paper/5866-pointer-networks.pdf>
- Wiengarten F, Singh PJ, Nazarpour FB, A (2017) Impact of mass customization on cost and flexibility performances: the role of social capital. *Oper Manag Res* 10(3):137–147. <https://doi.org/10.1007/s12063-017-0127-2>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.