2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA) | 978-1-7281-2989-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ETFA45728.2021.9613480

Recommending Assembly Work to Station Assignment Based on Historical Data

Ouijdane Guiza*, Christoph Mayr-Dorn†, Michael Mayrhofer*
Alexander Egyed† , Heinz Rieger‡, Frank Brandt§
* Pro2Future GmbH, † Johannes Kepler University, ‡ Wacker Neuson GmbH, § DMTM
*firstname.lastname@pro2future.at, †firstname.lastname@jku.at

***Abstract*—The Assembly Line Balancing Problem (ALBP) is of great relevance for manufacturing companies improving the line efficiency and productivity and thus maximizing production profits. Multiple exact, heuristic and meta-heuristic methods have been applied to solve the ALBP. These optimization methods consist in producing a feasible line balance, i.e. the partitioning of assembly tasks among available work stations based on, among others, the precedence graph. Such a graph describes the technological and organizational precedence constraints between tasks. Unfortunately, the assembly precedence relations, in the automotive and related industries for example, are often outdated, incomplete or altogether unavailable. This limits the applicability of the available approaches to real-world assembly systems. Grounded in an industry use-case, we propose a novel approach for the assistance in the upfront assignment of assembly tasks to stations. We recommend station assignments relying on historical data of prior feasible assembly balances of different products. We evaluate our approach against real industry data. On average, our approach is able to provide station assignment recommendations for 91% of the tasks at 82% precision.**

***Index Terms*—Assembly Line Balancing Problem, task assignment, assembly process, Jaccard similarity, assignment recommendation.**

I. Introduction

The thorough planning and configuration of assembly systems are of uttermost importance for manufacturing companies maximizing the production efficiency and profit [1]. The ALBP is the partitioning of assembly work among stations with respect to prioritized objectives [2]. These objectives are cost or profit oriented and aim to minimize the number of stations and/or maximize the line efficiency [3].

The Assembly Line Balancing Problem (ALBP) has drawn considerable attention from the academic and industrial communities for decades [2], [3], [4]. The formerly presented solutions, however, are not always feasible in real-world assembly systems. Boysen et al. draw attention to the gap between the effort invested in solving the ALBP within the research community and its applications in industrial settings [5]. Falkenauer further specifies the limited number of commercially available software related to the ALBP in the automotive and related industries, such as construction vehicles manufacturing [6]. They explain this by complex combinations of industrial requirements which are often addressed individually in research. Alternatively, applying the ALBP solutions is often unfeasible because of the lack of input data namely the assembly precedence graph. The assembly precedence graph is a directed acyclic graph describing the feasible order of execution of assembly work in the presence of restricting technological and organizational constraints [7]. This data is often outdated, incomplete or altogether unavailable. The creation and maintenance of the ever-changing precedence relations require extensive time and effort [8].

Klindworth et al. state the example of the automotive industry where experts rely on their tacit knowledge of precedence relations and other constraints to deliver a feasible assembly line balance [7]. Solving the ALBP includes setting the assembly systems capacity (cycle time, number of stations, number of workers, etc.), as well as properly assigning the assembly work to the corresponding stations [1]. The latter can only be attained through expert knowledge of the line's assembly activities and constraints. Experts often assign the assembly tasks manually based on their tacit knowledge of related activities that can be processed together in one environment (i.e., a station). For larger products for example, this results in having to process a large number of assembly tasks and identifying a preliminary station assignment. As the next step, the experts proceed to manually balance the pre-assigned tasks producing a feasible assembly balance. This step aims at balancing the work load assigned to stations and workers while maximizing the line efficiency, minimizing the number of stations and/or workers, etc.

The manual assignment and balancing is usually an iterative process to optimize the final balance while taking input from line leaders, workers, logistics department, etc. These iterations take considerable time, and thus determine how often it is economical to rebalance a line to better match demand (and supply of parts). A vital step in speeding up the balancing task, thus, is to produce a correct and usable preliminary assignment of assembly tasks to stations as the basis for the first iteration. This reduces the number of iterations needed. This paper addresses the problem of obtaining such a preliminary task assignment.

To this end, we propose an approach assisting in the upfront assignment of assembly tasks by providing station assignment recommendations without relying on the precedence relations. This approach is based on historical data from prior assembly balances of different products and derives station assignment information based on calculated similarities among tasks.

Specifically, our contributions in this paper are (i) an algo-

978-1-7281-2989-1/21/$31.00 ©2021 IEEE

TABLE I
SIMPLIFIED ASSEMBLY LINE BALANCE EXAMPLE OF AN EXCAVATOR.

Stations	Workers		Tasks
			Hydraulics mounting
Station1	W1	$T_{1.1}$	Mounting of the hydraulic oil tank
	W2	$T_{1.2}$	Mounting of the hydraulic pump
		$T_{1.3}$	Mounting of the hydraulic hoses
		$T_{1.4}$	Mounting of the hydraulic pedals
			Arm contruction
Station2	W3+W4	$T_{2.1}$	Mounting of the Boom
		$T_{2.2}$	Mounting of the Arm
		$T_{2.3}$	Mounting of the Bucket
			Cabin construction
Station3	W5	$T_{3.1}$	Mounting of the seats
		$T_{3.2}$	Mounting of the cabin frame
		$T_{3.3}$	Mounting of the Radio
		$T_{3.4}$	Mounting of the windows

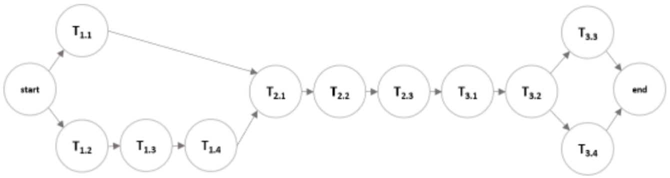

Fig. 1. Simplified precedence graph of an excavator assembly process.

rithm for calculating structural similarities between assembly tasks based on the task's most relevant activities and (ii) an assembly task to station assignment recommendation algorithm based on the calculated similarities. On average, our approach is able to provide station assignment recommendations for 91% of the tasks at 82% precision.

The remainder of this paper is organized as follows; We formally define the problem as well as relevant background information in Section II. In Section III, we discuss the former solutions presented in literature. Section IV introduces our approach for providing assembly work to station assignment recommendations. A thorough evaluation is then presented in Section V, before Section VI concludes the paper with an outlook on future work.

II. PROBLEM DEFINITION AND BACKGROUND

An assembly process P is a set of assembly tasks $T_i, i \in \{1, .., n\}$ (hereafter referred to as tasks) that are progressively executed to assemble the final product. Due to technological or organizational conditions, tasks are restrained with precedence constraints that ought not to be violated (e.g., one cannot mount an excavator's bucket before its boom). These constraints, defined in a directed acyclic precedence graph, describe the fact that some tasks have to be imperatively completed before other tasks. Tasks can thus be processed in parallel or in sequence in accordance with the precedence graph. Figure 1 illustrates an example of a precedence graph defined for a simplified assembly process of an excavator defined in Table I (other details of the excavator assembly balance from Table I will be presented later in this section).

Each task T_i is characterized by a predetermined task time t_i. Task times are often obtained using Predetermined Motion Time Systems (PMTS). Methods-Time Measurement (MTM) [9] is currently one of the most used methods in manufacturing. This method consists in the analysis of each task into basic human motions such as reach, grasp, move, etc. A composition of these human motions creates higher-level building blocks that we refer to as sub-tasks st. Subtasks define recurrent activity patterns that can be reused in building all assembly tasks for all products of the same or similar categories. Examples of such sub-tasks include "use crane to move heavy part for x meters", "screw in and tighten x screws with screwdriver or fork wrench", "mount electric cable". These sub-tasks describe basic yet generic assembly activities that it can build a wide range of assembly tasks. Each assembly task can thus be defined as $T_i = \{st_j; j \in \{1, .., m\}\}$

An assembly line A is composed of a set of l stations $S_k, k \in \{1, .., l\}$. Straight and U-shaped are the most commonly implemented assembly line layouts. One or multiple product models can be assembled on the same assembly line. The station load Σ_k is the set of tasks assigned to the station S_k. The station time is the time required to process all tasks assigned to the station. In the case of paced assembly lines, stations are tightly coupled with a common cycle time c. Every cycle c, the whole assembly line, thus all products under assembly, advances by one station. Each station time must not exceed the fixed cycle time c. A team of one or more workers, human or robots, is also assigned to each station thus responsible for the processing of its assigned tasks. The necessary tools and parts are available at each station.

ALBP is the partitioning of the assembly work among stations, optimized towards predetermined objectives [2]. We define the balance B as one feasible balance for process P on an assembly line A; $B = \{\Sigma_k; k \in \{1, .., l\}\}$. Table I defines an example of a simplified feasible assembly balance for an excavator. It details the tasks partitioning among stations and the workers assignment. Other aspects, such as the assembly parts, available tools, etc. are outside the scope of this paper.

ALBP has been extensively researched for decades. A wide ranging set of solutions has been proposed as a result. A well known classification of the problem was introduced by Baybars in [10]. The simplified version is a single model problem referred to as the the Simple Assembly Line Balancing Problem (SALBP). This category is restrained by several assumptions in order to simplify the problem. These assumptions however are often restricting and do not accurately portray the assembly lines in the real-world production systems. The relaxing of one or more of the assumptions results in the more realistic version denoted the General Assembly Line Balancing Problem (GALBP).

For the remainder of this paper we make the following assumptions inspired by the SALBP assumptions.

- Production of one homogeneous product in the presence of variants.
- Paced assembly line with a fixed common cycle time c.

- An assembly task $Ti, i \in \{1, .., n\}$ can not be split among stations.
- A pre-defined set of sub-tasks is used to build all assembly tasks.
- Data of prior assembly balances of different products (belonging to the same product category for example) are available.

III. Related Work

Since the first mathematical formulation of the ALBP by Salveson [11], several optimization models have been presented and discussed in literature [2]. In the effort to tighten the gap between literature and industry, extensions of the original SALBP were also studied and solved [3], [4]. This includes different assembly layouts, mixed and multiple model lines, non-deterministic task times, etc. The existing solutions apply exact methods [12], [13], [14], heuristic methods [15], [16], [17] and meta-heuristic methods [18], [19].

The existing approaches produce optimized assembly sequences with respect to specific objectives mainly using as input the precedence graph. However in real-world assembly systems, the intensive required manual input increasing with the number of tasks (up to several hundreds or even thousands) prevents manufacturers from collecting and maintaining precedence relations [20]. The knowledge of the precedence relations exists implicitly in the heads of experts, who manually generate feasible plans for the assembly line or segments of it [8].

To solve this impracticality, some approaches opt for the automatic or semi-automatic generation of the precedence relations. Niu et al. propose an approach for the generation of precedence graph based on a hierarchical relation graph and the mating relation graph, derived from the CAD model of the product under assembly [21]. Klindworth et al. present the Basic Learning Precedence Graph Concept (BLGC) based on past feasible sequences and expert interviews [7]. An extension is then introduced in [8] by integrating additional sources of data and enhanced interview guidelines. These approaches, although useful for product re-balancing, require a number of feasible sequences of the same product causing a cold-start problem. Other sources of data are required (such as CAD) yet not always available.

We present an approach that, based on historical data of prior balances of other products, computes similarities between tasks and concludes the station assignment. Very few assembly task similarity approaches have been presented in literature. As an example, we mention Renu and Mocko who investigated the use of different textual comparison measures to calculate assembly tasks similarities for knowledge retrieval and reuse [22].

IV. Approach

We propose the following approach to produce a preliminary assignment of tasks to stations based solely on historical data and without referring to a precedence graph. The input of the approach is a previous assembly balance of a different product designated by the reference balance B hereafter. $P = \{T_i; i \in \{1, .., n\}\}$ identifies the process to be assigned to an assembly line, while $P_B = \{T_p; p \in \{1, .., q\}\}$ describes the process corresponding to the reference balance B.

Algorithm 1 outlines the overall approach for task assignment. In short, for each task of P, its station assignment is inferred from its most similar task in B. A preliminary filtering of the tasks of B is executed to narrow the search possibilities (sub-section IV-A). We start by determining weights of the sub-tasks in P and B (Sub-section IV-B). These weights measure how "informative" a sub-task is when proceeding to calculate the similarities between tasks (sub-section IV-C). A dynamic threshold is also computed depending on the task's frequency (Sub-section IV-D). Once the most similar tasks are determined, a station mapping is applied to ensure the assignment to the corresponding assembly line station in case the reference balance assembly line layout differs from the layout used for the product to be balanced (Sub-section IV-E).

A. Meta-data Based Tasks Filtering

The default approach in determining the most similar task pairs is to conduct a pairwise comparison, resulting in $|P| * |B|$ comparisons; thus, an amount that increases exponentially with the number of tasks in P and B. Indeed, it is not necessary to make so many comparisons as, for example, tasks for testing certain functionality towards the end of the assembly line, do not need to be compared to tasks related to chassis assembly at the beginning of the line. Such coarse grained grouping cannot be determined based on the sub-task information. Hence we refer to task meta-data that is typically available. Examples include the information on the segment on the assembly line (pre-assembly, main assembly, testing, quality assurance, etc.) or the assembly categories (hydraulics, electronics, chassis, etc.). The preliminary tasks filtering reduces considerably the processing time.

Hence, the purpose of preliminary task filtering is to reduce the number of pairwise task similarity calculations as calculations occur only between tasks of the same grouping. Based on the task's T_i meta-data, several tasks of the reference balance B can be discarded as similar candidates and thus no similarity calculation is needed for the pair. Note that this step is not mandatory and can be skipped if such meta-data is not available. In this case, efficient data management structures for reusing similarity calculation and other heuristics can reduce the amount of necessary calculation. These, however, are outside the scope of this paper.

B. Tasks Weighting

The task similarity depends on the structure of each task, i.e. its constituent sub-tasks. However, not all sub-tasks are of comparable relevance. We assign to each sub-task a weight reflecting its relevance to the higher-lever task in the context of a given process. This weighting scheme is inspired by the term frequency-inverse document frequency (tf-idf) weighting factor used primarily in the information retrieval domain. The

Algorithm 1: Similarity-based Task to Station Assignment

```
Input: Process P, Balancing B
Output: A station assignment for each $T \in P$
for $T_i$ in $P$ do                                   ▷P sub-tasks weighting
    for $st$ in $T_i$ do
        determine weight of $st$
    end
end
for $T_j$ in $B$ do                                   ▷B sub-tasks weighting
    for $st$ in $T_j$ do
        determine weight of $st$
    end
end
recommendations = Listtask, station
for $T_i$ in $P$ do
    mostSimilarTask = {task:null, similarity:0}
    for $T_j$ in $B$ do
                                                      ▷Initial Task Filtering
        if metaDataInfo($T_i$) $\neq$ metaDataInfo($T_j$) then
            continue;                                 ▷Proceed with next reference task
        end
                                                      ▷Pairwise Task Similarity
        if similarity($T_i$,$T_j$) $>$ mostSimilarTask.sim then
            mostSimilarTask.task $\leftarrow T_j$
            mostSimilarTask.sim $\leftarrow sim(T_i, T_j)$
        end
    end
                                                      ▷Rare case: return empty recommendation
    if mostSimilarTask.task == null then
        recommendations.add($T_i$, null);
        continue;                                     ▷Proceed with next task
    end
                                                      ▷Dynamic threshold to improve precision
    $\tau$ = getThreshold(metaDataInfo($T_i$))
    if mostSimTask.sim $> \tau$ then
        S = getStation(mostSimTask.task)
                                                      ▷Station Mapping
        recStation = getStationMapping(S)
        recommendations.add($T_i$, recStation);
    else
        recommendations.add($T_i$, null);
    end
end
Return recommendations
```

basic idea is: the more often a sub-task occurs in the various tasks, the less useful it is for determining task similarity.

The weight of a sub-task st_j of a task T_i is calculated as follows as the product of two metrics.

$$w(st_j, T_i) = stf(st_j, T_i) * itf(st_j, P) \qquad \forall i \in \{1,..,n\}\forall j \in \{1,..,m\} \tag{1}$$

The first metric is sub-task frequency $stf(st_j, T_i)$. It measures how frequently st_j occurs within T_i, Eq. 2. It is calculated as the ratio of the raw count of the sub-task within the higher-level task $f(st_j, T_i)$ by m, the total number of sub-tasks in T_i.

$$stf(st_j, T_i) = \frac{f(st_j, T_i)}{m} \qquad \forall i \in \{1,..,n\}\forall j \in \{1,..,m\} \tag{2}$$

The second metric is inverse task frequency $itf(st_j, P)$, Eq. 3. This metric is inversely proportional to the number of tasks containing the sub-task st_j. It measures how frequently the sub-task is used throughout the process indicating its relevance. A less frequent sub-task, considered more relevant, results in a higher value of itf and thus a higher overall weight. Inversely, itf diminishes the weight of very frequent sub-tasks.

$$itf(st_j, P) = \log\left(\frac{n}{count(T \in P; st_j \in T)}\right) \qquad \forall j \in \{1,..,m\} \tag{3}$$

C. Task Similarity Calculation

Different methods can be used to measure the degree of similarities between assembly tasks. Textual similarity based on the task id or description or similarity of the used assembly parts are viable options. For this approach, we propose a structural assembly task similarity based on the tasks' constituent sub-tasks. To compute the similarity between two tasks T_i and T_j, we start by representing the two task sets as ρ-dimensional vectors V_i and V_j, where $\rho = |T_i \cup T_j|$. Several measures such as Jaccard's index, Sorensen–Dice coefficient, cosine index, and overlap coefficient have been used for data sets similarity calculations. For this approach, we use Jaccard's weighted similarity index[23], calculated as follows;

$$J(V_i, V_j) = \frac{\|V_i\|_1 + \|V_j\|_1 - \|V_i - V_j\|_1}{\|V_i\|_1 + \|V_j\|_1 + \|V_i - V_j\|_1} \tag{4}$$

We use the Jaccard's weighted similarity index with the sub-tasks' weights (as introduced in Subsection IV-B above). The reason for using weights is that very common sub-tasks should not result in a high similarity measure. Hence, for a pair of tasks, one sharing only three common sub-tasks, and another pair sharing only three rare sub-tasks, the former pair will yield a lower similarity score compared to the latter pair.

Specifically the similarity score is in range 0 to 1: $0 \leqslant J(V_i, V_j) \leqslant 1$. A value of 1 indicates a perfect similarity between the two tasks (i.e., both tasks consist exactly of the same sub-tasks) while a value of 0 indicates absolute dissimilarity (i.e., both tasks share no sub-tasks).

D. Dynamic threshold adjustment

By default, a task pair with a similarity value below the similarity constraint is disregarded. The setting of a suitable similarity threshold is not a trivial process requiring usually expert input. To simplify this process, we propose a dynamic threshold adjustment based on the task's meta-data. The idea is similar to obtaining the weight for a sub-task. For tasks belonging to larger grouping, a multitude of candidates is considered and thus a higher threshold is required to assure the selection of the most similar tasks. For groupings with a fewer number of tasks, already a low similarity score will be sufficient to obtain a useful station recommendation.

To this end, we define G as the group of tasks sharing the same meta-data (pre-assembly group, testing group, hydraulics mounting group, etc.). Then we assign weights to each group, Eq. 6. $w(G, P)$ is based on the inverse group metric measure $igf(G, P)$, Eq. 5. The inverse group metric works similarly to the inverse task frequency (Eq. 3). The larger the group (compared to the overall task count), the closer to zero igf becomes. Eq. 6 scales igf into the interval 0 to 1. Thus $w(G, P)$ is 0 for large groups, which require no additional lowering of the similarity threshold, and 1 for smaller groups, where a lower similarity threshold is acceptable to obtain higher coverage at similar precision. As the evaluation shows (SectionV), this achieves good coverage, without trading off precision.

Finally, the dynamic threshold $\tau(G)$ is derived from $w(G, P)$ and the base threshold τ_b, Eq. 7. The base threshold just needs to be roughly set in the interval 0 to 1, and is then automatically reduced with the group weight in the range of 0 to 0.5. Hence, those tasks that reside in small groups are subjected to a much smaller (i.e., up to 0.5 smaller) similarity threshold.

$$igf(G, P) = \log 2(\frac{n}{\|G\|}) \tag{5}$$

$$w(G, P) = \frac{max(min(igf, 10), 0)}{10} \tag{6}$$

$$\tau(G) = \tau_b - \frac{w(G, P)}{2} \tag{7}$$

E. Task Assignment Recommendation

Once the similar tasks are filtered based on the dynamic threshold, for each task in P the tasks with the highest similarity value in B are kept. Once the most similar tasks are identified, we can extract its assignee stations. If the process P and the reference balance B are balanced on the same assembly line, we recommend the same station. When the assembly line layout is not the same (e.g., additional stations, or fewer station available), a station mapping is necessary. We suggest the following heuristic for a linear assembly layout. Take the identified station's preceding station in B and check if it matches a station in P, if so, assign to the successor station in P. If not, take the identified station's successor station in B and check if it matches a station in P, if so assign to the predecessor station in P. In our prototype and evaluation, we limited such checking to on station prior or later (and gave no recommendation is case of no station match). This approach, however, can be easily extended to consider stations further away as well, or apply it to a network of stations and then consider stations in a particular hop distance.

V. Evaluation

A. Evaluation Data

We evaluate our approach based on real assembly data provided by our industry partner Wacker Neuson, a leading manufacturer of compact construction machines. The evaluation data is composed of a total of 16 assembly balances for 4 different excavator models (B_0 to B_{15}). Assembly balances covering 3 of these models (B_0 to B_{11}) belong to a similar excavator family whilst the remaining balances (B_{11} to B_{15}) describe the balancing of a model judged different than the others by domain experts (different frame design, smaller size, etc.). These balances were iteratively optimized by balancing experts at Wacker Neuson and have been deployed on the assembly line. For each model, 4 different assembly balances are available for distinct cycle times. Each assembly balance describes the tasks to station assignments, the number of workers needed and the workers assignments for the corresponding cycle time. For assembly balances of the same model, minor variances in the assembly process also occur (number of tasks, task renaming, tasks added etc.). Note that these minor variances are not due to different configurations of the same model since all processes describe the assembly of the product in all its possible variants (i.e., the 150% process). These minor differences are rather due to continuous small improvements.

The 16 processes corresponding to the assembly balances cover all possible product variants and contain a total number of tasks varying between 224 and 701. We eliminate redundant tasks describing supplementary assembly activities such as order reading, station booking, product transport, etc. These tasks are usually repeated in all stations and thus no station assignment recommendation is necessary. A total of 355 subtasks are used as the building blocks of all tasks of all products including the 4 excavator models included in this evaluation data.

The assembly balances describe then the assignment of these processes to different assembly lines. The assembly line are composed of stations of a total number varying between 11 and 22. Each station hosts an assembly worker team containing 1 to 7 workers responsible for the performing of its assigned tasks.

From each balance, we extract the task to station assignment data that will serve as our ground-truth data. We then proceed to apply our approach to generate station assignment recommendations for the corresponding process. As a first step, we focus on the balancing of assembly process using a reference balance of a similar model. We apply our approach on processes P_0 to P_{11} using a reference balance of the two remaining product balances (8 in total). For processes describing the first model for example, P_0 to P_3, balances B_4 to B_{11} are used as reference balances for the station assignment recommendation approach. Note that for a given process, we never use the assembly balances (of different cycle times) of the same model as we aim to evaluate the ability to balance a new model based on similar models.

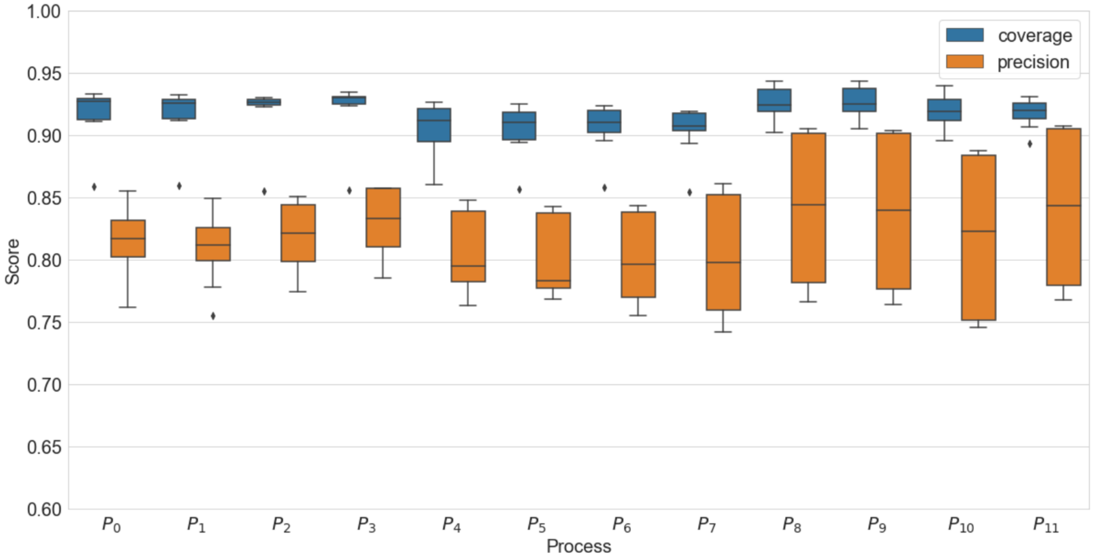

Fig. 2. Recommendation quality based on similar processes.

As a second step, we also evaluate our approach relying only on assembly balances of a dissimilar product. The purpose is to demonstrate the importance of choosing a sensible reference product, a typically straight forward task for a user familiar with the product portfolio. We evaluate the station assignment recommendations provided for the balancing of processes P_0 to P_{11} using B_{12} to B_{15} as reference balances and vice versa (assigning processes P_{12} to P_{15} using B_0 to B_{11} as reference balances)

We evaluate our station assignment recommendation approach with a base threshold τ_b of 0. This achieves that we recommend the tasks with the highest similarity independently of the threshold for the purpose of this evaluation. In an actual tool, the end user would select the desired trade-off between coverage and precision. In any case, the similarity measure and the dynamic threshold together provide insight into the recommendation to enable the user to judge the trustworthiness of a recommended assignment. To this end, we also evaluate our proposed dynamic threshold approach with comparison to standard static threshold.

The results of our evaluation and its interpretation is discussed in the following section.

B. Results and Discussion

To evaluate our task assignment recommendation approach, we calculate two metrics namely the coverage and precision. Figure 2 displays the calculated results for each of the 12 assembly processes (P_0 to P_{11}) showing the distribution of the two metrics (varying between 0 and 1) depending on the reference balance used.

The coverage, measuring the completeness of our approach, is calculated as the ratio of tasks our approach is able to provide a station assignment recommendation for, to the total number of tasks. The coverage values, as shown by the blue boxes in Figure 2, are relatively high and vary between 0.85 and 0.94. The average coverage values for each process are between 0.9 and 0.93. This means that on average, for the 12 processes of the evaluation data, our approach is able to provide station recommendations for 90% to 93% of all tasks.

We also measure precision of our approach calculated as the ratio of correct assignment recommendations to the total of provided recommendations. The calculated precision values vary between 0.74 and 0.91 for all the 12 processes of the evaluation data whereas the average precision varies between 0.8 and 0.84. This means that, on average, 80% to 84% of the station assignment recommendations provided by our approach are indeed correct. Note, however, that these results are obtained when using no similarity threshold. Hence, by selecting a suitable reference balancing and by increasing the threshold, one can obtain highly precise recommendations.

We also evaluate our approach in the absence of reference balances of similar products. First we apply our approach to assist in the upfront assignments of processes P_0 through P_{11}, using as reference balances B_{12} to B_{15}. The results are displayed in Figure 3. The coverage values vary between 0.33 and 0.67 while the precision values vary between 0.22 and 0.5. Figure 4 displays the coverage and precision scores for the station assignment recommendation approach applied for processes P_{12} through P_{15}, using as reference balances B_0 to B_{11}. The coverage varies between 0.67 and 0.83 while the precision values are between 0.2 and 0.8.

The results displayed in Figure 2 are calculated without limiting the approach with a similarity threshold. This demonstrates the quality of our approach, even without using a threshold and thus maximizing the amount of recommendations, hence lowering the remaining effort for the user of assigning the tasks without a recommendation. The (dynamic) threshold still has important value for the user, as he/she may set the threshold value to adjust the set of recommendations,

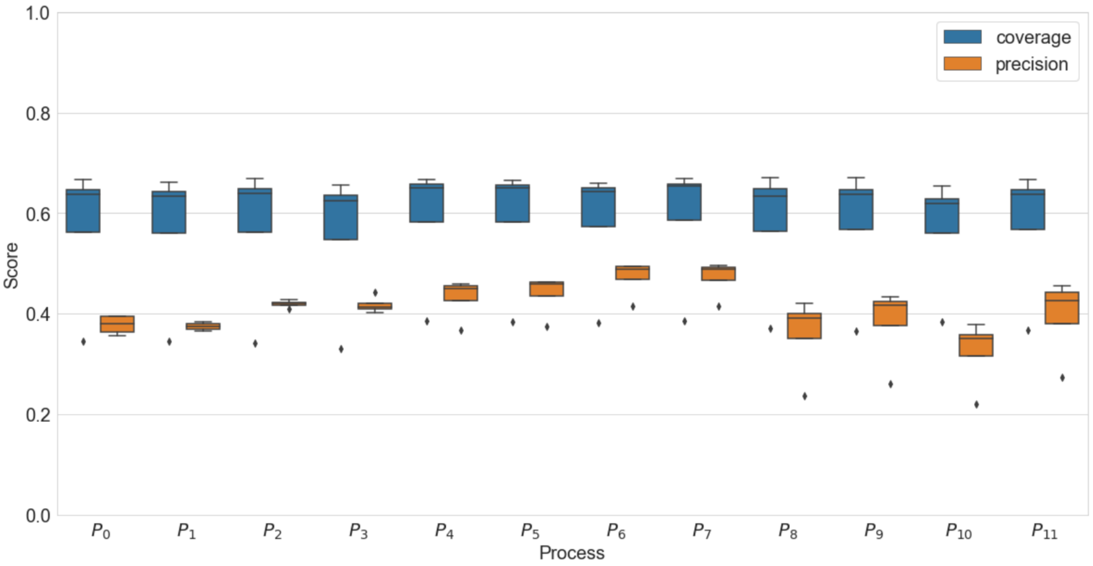

Fig. 3. Recommendations quality based on dissimilar processes.

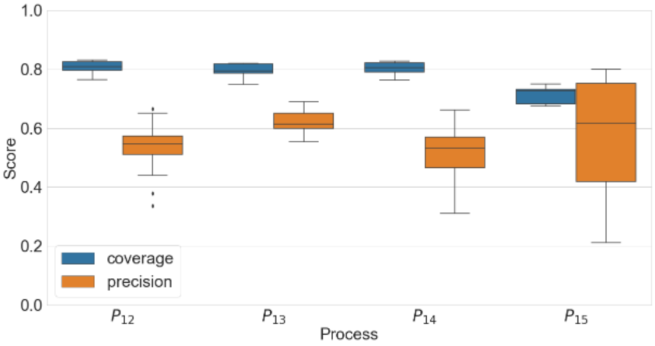

Fig. 4. The four dissimilar processes recommended on the other 12 similar ones.

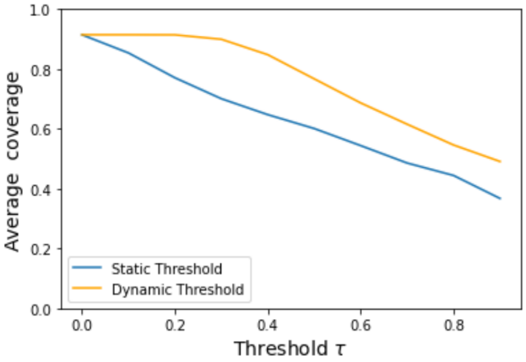

Fig. 5. Comparison of coverage at different threshold levels.

either by disregarding too low similarity recommendations all together or by primarily inspecting the correctness of the low similarity recommendations. Especially in cases, where a higher precision is needed (in the case of novice users, or when no meta-data is available for the preliminary task filtering step) setting a threshold is useful to filter the recommendations derived from lower calculated similarities.

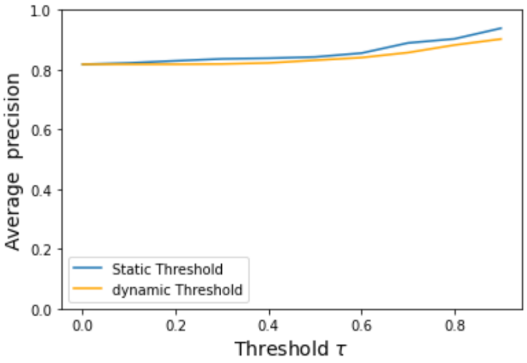

Fig. 6. Comparison of precision at different threshold levels.

To this end, we evaluate our dynamic threshold approach compared to a standard static threshold. Figure 5 and Figure 6 compare the obtained coverage, respectively precision, at different threshold values averaged over the 12 similar products. One notices that while the precision is slightly lower for the dynamic threshold, coverage is much higher (hence lowering the work for the user without introducing many incorrect recommendations). The figures also show that a tuning of the threshold below 0.4 has virtually no effect on coverage, and only thresholds above 0.6 tend to increase precision. Note, as explained above, that choosing a suitable candidate balance for comparison will increase the precision numbers even more. Automatically determining a heuristic for determining the near-optimal candidate reference balance, however, is outside the scope of this paper.

VI. Conclusion

This paper presented a novel approach for the assistance in the phase of upfront task to station assignment in the context of assembly line balancing. Current solutions to the ALBP rely mainly on the precedence graph to optimize an assembly balance with respect to predefined objectives. In reality however, the precedence relations are often outdated, incomplete, or altogether unavailable. Alternatively, experts rely on their tacit knowledge to produce a feasible assembly balance assigning tasks to stations and at a later stage balancing the load to meet the predefined objectives with respect to the implicitly known precedence relations. Our approach provides assistance to the balancing experts in the phase of the upfront task to station assignment thereby reducing the number of iterations and hence time to obtain a final balance. Evaluated against real industry data, our approach is able to provide station assignment recommendations for 91% of the total of tasks with a precision of 82%. We also presented a dynamic threshold approach's that improves the approach coverage as compared to standard static threshold. Future work includes the investigation of combining multiple reference balances while providing a heuristic suggesting to the user the most suitable candidate to use as a reference balance to yield the best results. We are preparing a user study with balancing experts from industry to determine the usability and usefulness of a prototype implementing our approach.

Acknowledgment

This work has been supported by the FFG, Contract No. 881844: "$Pro^2Future$ is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG."

We would also like to thank our company partners Wacker Neuson Linz GmbH and Deutsche MTM-Gesellschaft Industrie- und Wirtschaftsberatung mbH for their assistance and useful input.

References

[1] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," *European Journal of Operational Research*, vol. 183, pp. 674–693, 12 2007.

[2] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *European Journal of Operational Research*, vol. 168, pp. 666–693, 02 2006.

[3] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *European Journal of Operational Research*, vol. 168, pp. 694–715, 02 2006.

[4] P. Sivasankaran and P. Shahabudeen, "Literature review of assembly line balancing problems," *The International Journal of Advanced Manufacturing Technology*, vol. 73, pp. 1665–1694, 08 2014.

[5] N. Boysen, M. Fliedner, and A. Scholl, "Assembly line balancing: Which model to use when?" *International Journal of Production Economics*, vol. 111, pp. 509–528, 02 2008.

[6] E. Falkenauer, "Line balancing in the real world," *International Conference on Product Lifecycle Management*, 01 2005.

[7] H. Klindworth, C. Otto, and A. Scholl, "On a learning precedence graph concept for the automotive industry," *European Journal of Operational Research*, vol. 217, pp. 259–269, 03 2012.

[8] C. Otto and A. Otto, "Multiple-source learning precedence graph concept for the automotive industry," *European Journal of Operational Research*, vol. 234, p. 253–265, 04 2014.

[9] H. B. Maynard, G. J. Stegemerten, and J. L. Schwab, "Methods-time measurement." 1948.

[10] I. Baybars, "A survey of exact algorithms for the simple assembly line balancing problem," *Management Science*, vol. 32, pp. 909–932, 08 1986.

[11] M. Salveson, "The assembly line balancing problem," *Journal of Industrial Engineering*, vol. 6, pp. 18–25, 01 1955.

[12] Z. Li, I. Kucukkoc, and Q. Tang, "A comparative study of exact methods for the simple assembly line balancing problem," *Soft Computing*, vol. 24, pp. 1–17, 08 2020.

[13] M. Vilà and J. Pereira, "A branch-and-bound algorithm for assembly line worker assignment and balancing problems," *Computers and Operations Research*, vol. 44, p. 105–114, 04 2014.

[14] D. Morrison, E. Sewell, and S. Jacobson, "An application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset," *European Journal of Operational Research*, vol. 236, p. 403–409, 07 2014.

[15] L. Borba, M. Ritt, and M. Cristóbal, "Exact and heuristic methods for solving the robotic assembly line balancing problem," *European Journal of Operational Research*, vol. 270, 03 2018.

[16] T. Hoffmann, "Assembly line balancing with a precedence matrix," *Management Science*, vol. 9, pp. 551–562, 07 1963.

[17] K. Fleszar and K. Hindi, "An enumerative heuristic and reduction methods for the assembly line balancing problem," *European Journal of Operational Research*, vol. 145, pp. 606–620, 02 2003.

[18] S. Ozmehmet Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *Journal of Intelligent Manufacturing*, vol. 19, pp. 49–69, 02 2008.

[19] S. Lapierre, A. Ruiz, and P. Soriano, "Balancing assembly lines with tabu search," *European Journal of Operational Research*, vol. 168, pp. 826–837, 02 2006.

[20] E.-D. Ammer, *Rechnerunterstützte Planung von Montageablaufstrukturen für Erzeugnisse der Serienfertigung*, 01 1985.

[21] X. Niu, H. Ding, and Y. Xiong, "A hierarchical approach to generating precedence graphs for assembly planning," *International Journal of Machine Tools and Manufacture*, vol. 43, pp. 1473–1486, 11 2003.

[22] R. Renu and G. Mocko, "Computing similarity of text-based assembly processes for knowledge retrieval and reuse," *Journal of Manufacturing Systems*, vol. 39, pp. 101–110, 04 2016.

[23] S. Ioffe, "Improved consistent sampling, weighted minhash and l1 sketching," 12 2010, pp. 246–255.