Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 712–717

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Offline segmentation of spatio-temporal order trajectories by mixed-integer linear programming for determining process times in production systems

Maximilian Volk[a,b], Carina Mieth*[,b,c]

[a]Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany
[b]TRUMPF Machine Tools Inc., Johann-Maus-Straße 2, 71254 Ditzingen, Germany
[c]TU Dortmund University, Emil-Figge-Straße 61, 44227 Dortmund, Germany

* Corresponding author. Tel.: +49 151 68831558. *E-mail address:* carina.mieth@trumpf.com

Abstract

Real-time indoor positioning systems in manufacturing systems are used to track production orders. This generates spatio-temporal trajectories which can be segmented to determine process times. We present formulations of the offline segmentation problem as mixed-integer linear programs (MILPs) that utilize the sequence of processing steps from ERP systems. The MILP formulations are compared with online heuristics in terms of their accuracy and computational effort on data generated with features from a real job shop. We show that in terms of accuracy our offline segmentation formulations outperform the online heuristics with increasing measurement errors, justifying their higher computational effort.

Keywords: indoor positiong systems; mixed-integer linear programming; process time estimation; trajectory segmentation; spatio-temporal production data

1. Introduction

Real-time indoor positioning systems in manufacturing systems provide valuable spatio-temporal data which can be used for various data-based services and advanced analytics [1]. For accurate production planning control it is crucial to have accurate knowledge about process and lead times of manufactured products. To date, the vast majority of these times have been relied upon to be correctly reported back by employees. However, there are large discrepancies between the actual and the reported times because workers unintentionally or intentionally start or finish jobs at the wrong times. This is due to the fact that reporting is still cumbersome or involves walking, so that, for example, collected reporting is only done at the end of the shift.

In the course of digitalization, indoor positioning systems (IPS) have become increasingly popular for tracking production orders with mobile sensors. This generates spatio-temporal data x_t which can be used to estimate lead times and thus, is the basis for automating the time clocking. For this, the spatio-temporal order trajectories need to be segmented to estimate process and lead times [2]. A challenge with this trajectory segmentation is that position measurements in production environments are prone to error due to shading and signal reflection [3].

We present different formulations of the offline segmentation of spatio-temporal order trajectories as mixed-integer linear programs (MILPs) that utilize the sequence of processing steps known for each product. The MILP formulations are compared with simple online heuristics in terms of their accuracy and computational effort on data generated with features from a real-world system. The studied system consists of a job shop with nine stations in a quadratic matrix layout with intersecting order trajectories, measurement errors, off-center geofences and occasional unexpected waiting times between processing steps.

2. Related Work

The literature contains a number of papers dealing with the prediction of lead times for products in the make-to-order manufacturing sector [4, 5, 6, 7, 8], some of them also consider smaller time proportions as cycle times [7, 9], processing times [8] or even shorter activities [10]. The main aim is to use these predicted times to improve production planning, but also the use in simulations is mentioned in [2, 7, 9]. Most often,

10.1016/j.procir.2022.05.050

data from a manufacturing execution system (MES) [5, 6, 7] is used for prediction. In [5], simple iterative algorithms are implemented for lead time prediction, whereas Pfeiffer et al. [7] use a filtering algorithm. Öztürk et al. [4] use synthetic data from a factory simulation model to predict lead times using data mining. In [6], simulated data is used to test the presented method that is based on multivariate regression. Mucientes et al. [8] use machine data in combination with information such as dimensions from technical drawings of historical products (wood-furniture industry) to create a regression function for each product type and machine, where the model is based on fuzzy rules. Our work, on the other hand, focuses on determining historically correct process times for each station. For this, we propose a classification problem that uses data from an ultra-wide band (UWB) based indoor localization system which is combined with information on the sequence of process steps needed to create each product.

In recent work [9, 10, 11, 12, 13], location data is used to derive information about the production system. Work from [11] and [12] show, that unsupervised methods can reveal basic information on the system layout e.g. storage locations and transportation routes. In [13], radio frequency identification (RFID) is used to compute better hypotheses for the position of objects with the help of online sequential extreme learning machines. The position hypotheses are then feed back into the real-time MES. Further analyses are made possible, but are not discussed or carried out. In [10], a sensor fusion system combining UWB localization data and sensor signals from fixtures is presented. The latter are fixtures at each work station within the considered production line of a harness manufacturing process. Every work piece needs to be clampeed at manual working station in order to be processed. This provides very reliable data on when a processing step has been performed. Unfortunately, such fixtures cannot be used in job shop manufacturing due to the inherently high variety of products. In [9], an automotive use case is discussed where data from a RFID system was analyzed. Since the considered data is waypoint-discrete, the data set already contains the assignment of workpiece to stations for each time point. In our paper, we start one step earlier, where we determine the assignment of the workpieces to the stations. We built upon the real-time indoor localization framework proposed by Mieth et al. [14] and previous work on semantic enrichment of spatio-temporal production data [2].

3. Data

The data generated by a production order tracked with an IPS consists of two-dimensional position measurements x_t at discrete time points t from a set $T = \{t_i \mid i \in I\}$ where $I = \{0, ..., i_{max}\}$ is the corresponding index set. On the shopfloor, there are objects and obstacles like machines or storage areas with which an order interacts by being stored or relocated. These objects are identified via their locations $g \in G$, where G is the set of all such locations. Each location g has a Point of Interest (POI) with position q_g and a polygonal area of interest (AOI), also called geofence, defined based upon the shopfloor layout. At each timepoint $t \in T$, using the position measurement x_t from the IPS, the location $g \in G$, in whose area of interest the order is located, can be determined. If uncertainty around the position is quanitified, for example in the form of a spatial confidence interval, the probability for an order in area of interest of location g at timepoint t is given by $p_{tg} \in [0, 1]$. These probabilities can be collected in the matrix $\mathbf{P} = (p_{tg}) \in \mathbb{R}^{|T|\times|G|}$. If information on uncertainty cannot be collected, we model $\mathbf{P} = (p_{tg}) \in \mathbb{B}^{|T|\times|G|}$, where $\mathbb{B}$ denotes the binary set $\{0, 1\}$.

The set of processing steps any tracked order can pass through in the specific production system will be denoted as K. For a single order, a process graph (V, E) can be obtained, for example from an ERP or MES system. Here, $V \subseteq K$ is the set of processing steps and $E \subseteq V \times V$ is the set of directed edges that the order takes between the process steps. The edge set $E^* = E \cup \{(v, v) \mid v \in V\}$ augments E with edges that lead from each vertex to itself. Thus, in between two timesteps the order either moves to the next processing step on its process graph or can stay at its current vertex. The adjacency matrix of the directed augmented process graph (V, E^*) will be denoted as $\mathbf{A} = (a_{k_1k_2}) \in \mathbb{B}^{|V|\times|V|}$. While most process steps in K are assigned one location in G and vice versa, there are process steps that can take place in multiple locations, e.g. a visual quality control or manual deburring. Moreover, there are locations like manual workplaces in which multiple processing steps can be performed. Differentiating between the sequential notion of a processing step and the spatial notion of a location yields the flexibility needed for the described cases. The assignment between process steps k and locations g specific to a production system is encoded in $\mathbf{H} = (h_{kg}) \in \mathbb{B}^{|K|\times|G|}$.

4. Estimating process times

At each timepoint $t \in T$, a classification of the order into one of the process steps $k \in K$ is encoded in the binary decision variables b_{tk}, collected in the matrix $\mathbf{B} = (b_{tk}) \in \mathbb{B}^{|T|\times|K|}$.

The order must be in exactly one process step at each timepoint. This can be ensured by a constraint that checks $\forall\, t \in T$ that $\sum_{k\in K} b_{tk} = 1$. Given such classifications, the process time $PT(k)$ for process step k can be estimated by:

$$\hat{PT}(k) = \frac{1}{2} b_{t_0k}(t_1 - t_0) + \sum_{i=1}^{i_{max}-1} \frac{1}{2} b_{t_ik}(t_{i+1} - t_{i-1}) + \frac{1}{2} b_{t_{i_{max}}k}(t_{i_{max}} - t_{i_{max}-1})$$

The objective is to determine the classifications b_{tk} such that the estimated process time $\hat{PT}(k)$ is as close to the true process time $PT(k)$ for each process step $k \in K$. For an order passing through processing steps $V \subseteq K$ the Mean Relative Absolute Error (MRAE) per processing step is

$$\text{MRAE} = \frac{1}{|V|} \sum_{k\in V} \frac{|\hat{PT}(k) - PT(k)|}{PT(k)}.$$

It will be used as the metric for evaluating and comparing methods to determine the classifications b_{tk}. These methods will be discussed in the following section.

5. Classification problems for trajectory segmentation

Two simple approaches to determine the classifications b_{tk}, discussed e.g. in [2], are Area-of-Interest (AOI) and Point-of-Interest (POI). They do not need any information on production processes. The first approach assigns the process step at each timepoint based on the location in whose area of interest the position is measured and the second assigns the process step with the closest point-of-interest q_g to x_t. In these naive approaches, the classification is only based on the current position measurement which could be faulty. This can be countered to some degree using filters and smoothing as discussed in [2]. By the law of large numbers one could argue that filters or the averaging over many measurements would neutralize measurement errors leading to asymptotically correct times. In practice however, three challenges arise: (1) While the tracked object is moving, only one measurement per position exists. (2) Due to battery limitations, IPS use sleep modes and do not provide observations while resting for longer periods of time. (3) Due to reflections and shading, measurement errors might not even be evenly or symmetrically distributed, leading to a biased position estimate. This is of particular importance in industry with big machinery or (sheet) metal objects on the shop floor.

Here, the classifications b_{tk} are determined by solving an optimization problem. This enables firstly to deviate in some classifications from the raw position measurements to lower the overall classification error and secondly to incorporate information about the sequence of process steps in the form of constraints.

The following mixed-integer linear optimization problem, will be referred to as CP_D. The optimization problem CP_D is prototypical, since the matrix of distances $\mathbf{D} = (d_{tg})$ as a parameter is specified later. Different distance notions that can be used will be discussed in Subsection 5.1.

In the objective function (1) of CP_D the inner sum is the classification error at each time t where d_{tg} is the distance between the measured and classified location g at time t. Therefore, the sum over the individual classification errors is to be minimized. The binary decision variable c_{tg} encodes the classification of the position measurement into a location g and vice versa. At each timepoint, the order is classified into exactly one of the locations in G (2). Similarly, a classification into exactly one process step is required (3). The order can be classified into being at location g only if the processing step k in which it currently is, can take place in that location (4). Under the assumption that the process graph (V, E) of the order is given, the process sequence can be integrated into the classification problem as presented in [2]. For this, here a linear constraint based on the adjacency matrix $\mathbf{A}$ is used. The sequence of processes is not violated only if $b_{t_i k_1} = 1 \wedge b_{t_{i+1} k_2} = 1 \Rightarrow (k_1, k_2) \in E^*$ which is equivalent to the linear constraint (5). Moreover, each vertex of the process graph has to be visited at least once to have processing times for each process step (6).

$$CP_D : \min \sum_{t \in T} \sum_{g \in G} c_{tg} d_{tg} \tag{1}$$

$$\text{s.t.} \sum_{g \in G} c_{tg} = 1 \qquad \forall t \in T \tag{2}$$

$$\sum_{k \in V} b_{tk} = 1 \qquad \forall t \in T \tag{3}$$

$$c_{tg} \leq \sum_{k \in V} b_{tk} h_{kg} \qquad \forall t \in T,\ \forall g \in G \tag{4}$$

$$b_{t_i k_1} + b_{t_{i+1} k_2} \leq 1 + a_{k_1 k_2} \qquad \forall k_1, k_2 \in V,\ \forall i \in I \backslash \{i_{max}\} \tag{5}$$

$$\sum_{t \in T} b_{tk} \geq 1 \qquad \forall k \in V \tag{6}$$

$$e \in \mathbb{R}^{|T|},\ b \in \mathbb{B}^{|T| \times |V|},\ c \in \mathbb{B}^{|T| \times |G|} \tag{7}$$

In case the order-specific process graph is not known, global process rules that all orders must follow in the specific production process or industry the indoor positioning system is working in, can be incorporated by modifying $\mathbf{A}$ in (5). While they are not as exact as constraints inferred from one specific order's process graph, they still can improve the classification accuracy compared to including no process knowledge.

The process graph can be used to reduce the number of considered locations by leaving out those that can never occur because no process step in V is assigned to them. For this, G is replaced by the set of possible geofences, G_s, which contains those locations $g \in G$ for which there exists a process step $k \in V$ such that $h_{kg} = 1$. If orders are diverse in their process graphs and each only use a small fraction of all locations on the shopfloor, this reduction in model size can yield a substantial decrease in model building and solving time which is advantageous for any productive use.

5.1. Distance measures

One choice for the distance matrix D is considering the euclidean distance between the position measurement x_t and the posisiton of the Point-Of-Interest q_g of location g. The resulting optimization problem is referred to as CP_{xPOI}. However, this notion of distance can be expected to have some flaws, since firstly the classification is highly sensitive to the position of the points of interest and the euclidean distance cannot account for differently sized areas of interests around them as stated in [2].

The flaw of the euclidean distance to the points of interest due to differently sized areas of interest can be avoided by considering the distance of the current measurement to the area of interest. Setting d_{tg} as the shortest distance from x_t to the outline of the polygonal area of interest of location g in CP_D yields the corresponding optimization problem which will be referred

to as CP_{xGeo}. If x_t is inside the polygon, this distance is set to zero.

Again considering the case that the geofences are polygons, a distance between two geofences can be calculated easily. The distance between two polygons is the minimum distance between two points on the polygon path. For the matrix of pairwise distances $\tilde{D} = (\tilde{d}_{g_1 g_2}) \in \mathbb{R}^{|G|\times|G|}$, 1 is added to every distance between two polygons to get a strictly positive distance also between bordering polygons and thus an error if a misclassification happens to a bordering area of interest. The distance between the current measurement and a location g_1 is then defined as $d_{tg_1} = \sum_{g_2 \in G} \tilde{d}_{g_1 g_2} p_{tg_2}$. If $\mathbf{P}$ is non-binary and given as the probability for a certain measurement to be in geofence g_2, d_{tg_1} becomes a probability-weighted distance measure.

Alternatively, one could disregard the distances between geofences and just differentiate between classification into a correct or incorrect geofence which was presented in [2]. In our framework, this would correspond to choosing $\tilde{d}_{g_1 g_2}$ to be 0 if $g_1 = g_2$ and 1 in all other cases. When $\mathbf{P}$ is binary this approach leads to a pure binary linear optimization problem which we call here CP_{none}.

5.2. Incorporating transports

All of the above models at a given time t only classify into one of the process steps and do not capture transport times explicitly, because transport times from a process step to the next are implicitly added to one or both of these process steps. Therefore, the duration of the individual process steps is systematically overestimated and this error grows with the ratio between transport and processing times. If the former are particularly high, e.g. because of waiting times between process steps that are not captured in the process graph, one needs to account for the transports in the model.

Transports are modelled with a modified process graph. For each transport, an artificial processing step is created. Then, connections to and from the artificial process step are created as edges of the new graph. For each edge $(k_1, k_2) \in E$, there has to happen a transport in between. Note that this is not the case for the edges in $E^* \backslash E$ as in those, the marker stays at the same station. Thus the set of vertices is augmented with the existing edges: $\tilde{V} = V \cup E$. The modified set of edges $\tilde{E}$ connects the modified vertices that is, one initial vertex with one initial edge or one initial edge with one initial vertex respectively. Constructing $\tilde{E}^*$ from $\tilde{E}^*$ analogously to E^* from E, yields the augmented process graph $(\tilde{V}, \tilde{E}^*)$ whose adjacency matrix is $\tilde{\mathbf{A}} = (a_{\tilde{k}_1 \tilde{k}_2}) \in \mathbb{R}^{|\tilde{V}|\times|\tilde{V}|}$.

Since the transports are treated as additional process steps, the assignment between process steps and locations $\mathbf{H}$ needs to be adjusted. We define $\tilde{K} = K \cup (K \times K)$ and $\tilde{\mathbf{H}} = (\tilde{h}_{kg}) \in \mathbb{B}^{|\tilde{K}|\times|G|}$ which is extended from $\mathbf{H}$ by assigning to all $k \in K \times K$, $\tilde{h}_{kg}$ the transport location g_T. It is the space that remains on the shopfloor after removing all other areas of interest.

After replacing V with $\tilde{V}$, A with $\tilde{A}$ and H with $\tilde{H}$ in CP_D which yields the optimization problem CP_{TD}, the redefinition of the distance d_{tg} remains. The optimization problem resulting from considering the distance between position measurement and geofence will be referred to as CP_{TxGeo}. If all non-transport geofences are polygons, the resulting transport geofence is a polygon as well albeit with holes. Setting d_{tg} to be the distance to the area of interest of location g that in case of the transport geofence, if the object is in another location, the distance will be measured within the corresponding hole. The optimization problem resulting from considering the distance between classified and measured geofence will be referred to as $CP_{TGeoGeo}$. Here, we define $\tilde{d}_{g_1 g_2}$ again by adding 1 to the polygon distance to get a strictly positive distance between bordering polygons, yielding an error if a misclassification happens to a bordering area of interest. This is particularly important here, since the transport area of interest borders that of any other process step. The distance between the current measurement and a location g_1 is defined analogous to the non-transport case. The transport equivalent of the binary-distance-based CP_{none} will be referred to as CP_{Tnone}. An adaptation of CP_{xPOI} to the transport case is not possible, since without additional assumptions on the transport paths there is no single POI for the transport process to which a distance could be defined.

6. Experiments and Results

The hypothetical production system for the evaluation is a job shop with nine process steps oriented in a quadratic matrix layout. The locations are indexed by their horizontal and vertical position relative to the bottom left process step (1,1). Each process step has a location with area of interest and point of interest associated with it. The areas of interest are quadratic and have a side length of 8m and feature a corridor of 4m between one another. Their point of interest is positioned centrally.

The considered order follows the sequence of process steps [(1,1),(3,2),(1,3),(3,1),(2,1),(2,2),(1,2),(2,3)] with a speed of 1 m/s. The trajectory with an offset error is depicted on the left side in Figure 1. At a given processing step, the order remains stationary for the duration of the process time, with an offset relative to the center of the area of interest. Offset waiting positions account for misspecification of areas or points of interest as well as for large asymmetric measurement errors that result from reflections or shading. On the way from process (1,3) to (3,1) there is an unexpected waiting position in the corridor with a certain waiting time.

A two-dimensional error drawn from a t-distribution with 3 degrees of freedom and standard deviation $SCALE_M$ is added to the position measurements. The larger kurtosis of the t-distribution leads to more large errors than a normal distribution with the same standard deviation which is closer to measurement errors observed in practice and enhances the robustness of the analysis. The position measurements and timestamps form a synthetic trajectory, examples of which are shown in Figure 1.

$SCALE_M$ is varied in [0 m, 0.5 m, 1 m, 2 m] and the process time (PT) is varied between 18 or 180 min for all process steps or 18 / 180 min alternating for each experiment. Finally, the corridor waiting time (CWT) is varied between zero

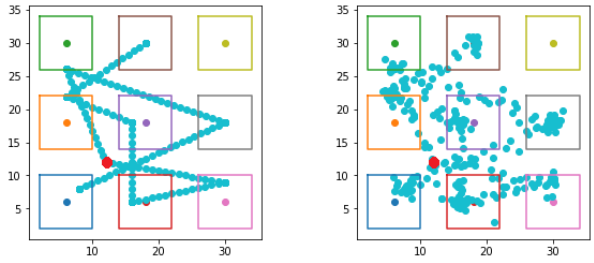

Fig. 1. Trajectories using offset process locations, waiting location (•) and t-distributed measurement errors with $SCALE_M = 0$ (left) and $SCALE_M = 1$ (right).

and 18 min. For each method and configuration, ten replications are performed, since the measurement errors are stochastic. The results in terms of solution time and accuracy are shown in Figure 2.

All computations are performed using *Python 3.7*. Distances between two polygonal areas of interest or a point and a polygonal area of interest are calculated with *shapely* [15]. The classification problems are solved using the *CP-SAT solver* from *Google OR-Tools* [16] which has access to 8 threads on a quad-core Intel i7 CPU for 150 seconds per problem. If the optimality of a found solution is not proven within the time limit, the best feasible solution known is used for evaluation. Since *CP-SAT Solver* is a pure integer programming solver, the distances are rounded to two digits.

6.1. Discussion of results

When there are no errors present the AOI and CP_{TD} perform best across all time configurations. However, the AOI method has an MRAE upwards of 10% for all configurations as soon as any measurement errors are present. As long as errors are small and the ratio between the unexpected waiting time and the process times is low, the POI method is working well. In these cases, there is no need for more complex models. As soon as the error standard deviation grows above 1m, the POI method becomes very volatile across replications and reaches a MRAE of 10% as well. In cases where the ratio of waiting time to process time is low , even if the errors are very large, the classification problems CP_D perform well with MRAEs of order 10^{-2}. If the waiting time becomes large relative to the process time, the POI method again becomes unreliable. The non-transport models CP_D however are also unreliable due to the systematic error mentioned in Section 5.2. In these cases, if there are no errors, the AOI method works well; as do the CP_{TD} transport models. For low, non-zero errors AOI also becomes unreliable, where the transport models can work well in some replications.

For error standard deviations above 1m and a high waiting time, all models perform badly, while at least for $SCALE_M = 1$ in some replications the transport models perform well. In the mixed process time case with waiting times, CP_{xGeo} and CP_{xPOI} are less robust with increasing errors than CP_{none} and CP_{GeoGeo}. In the majority of replications the MRAE of the latter two models is of order 10^{-2} with some worse outliers.

6.2. Recommendations

Based on the results of the numerical experiments recommendations on which methods to choose in different configurations of process and waiting times as well as measurement errors are given in Table 1.

Table 1. Recommendation of methods depending on time configuration and measurement errors.

Ratio waiting times/process times	Measurement errors	Recommendation
small	none	AOI
small	small	POI
small	large	CP_D
large	none/small	CP_{TD}

In the case of a large ratio of waiting to process times and large measurement errors none of the discussed methods can be recommended. The former can be reduced by introducing additional process steps in the orders process graph that model frequently occuring and long waiting times. Using fitting areas of interest, the non-transport models CP_D can be successfully applied as only large measurement errors remain. The latter can be reduced to below 0.5m by choosing a more precise IPS or increasing area coverage. Furthermore, context information like trustworthy classifications from RFID or other sensors can enrich our approach in the form of an additional constraints on b_{tk}. A suiting sensor fusion concept is described in [2].

7. Conclusion

A mixed-integer linear programming-based framework to segment spatio-temporal order trajectories was presented. With it, process times can be estimated that incorporate quantified measurement uncertainty and order-specific sequences. We showed that even under large measurement errors our offline segmentation formulations remain reliable and outperform the online heuristics in terms of accuracy, as long as unexpected transport times are relatively small. In the case that transport times are large, the tranport-based formulations can offer an advantage as long as measurement errors are small and enough computational resources are available. Comprehensive recommendations for practicioners on choosing the best method for process time estimation based on measurement error size and amount of transport times relative to process time are given. For the most difficult environment conditions, solutions are provided to enhance reliability of the methods.

Accurate estimation of historical process times is the basis for improved prediction of future process times. Thus, our work enables different use cases from production planning to simulation input modeling. In future work, global process knowledge that is common within a specific industry can be included to further enhance estimation accuracy, even in scenarios where no process graph for individual products is known. Additional potential lies in the inclusion of measurements from multiple sensors and tracking systems.

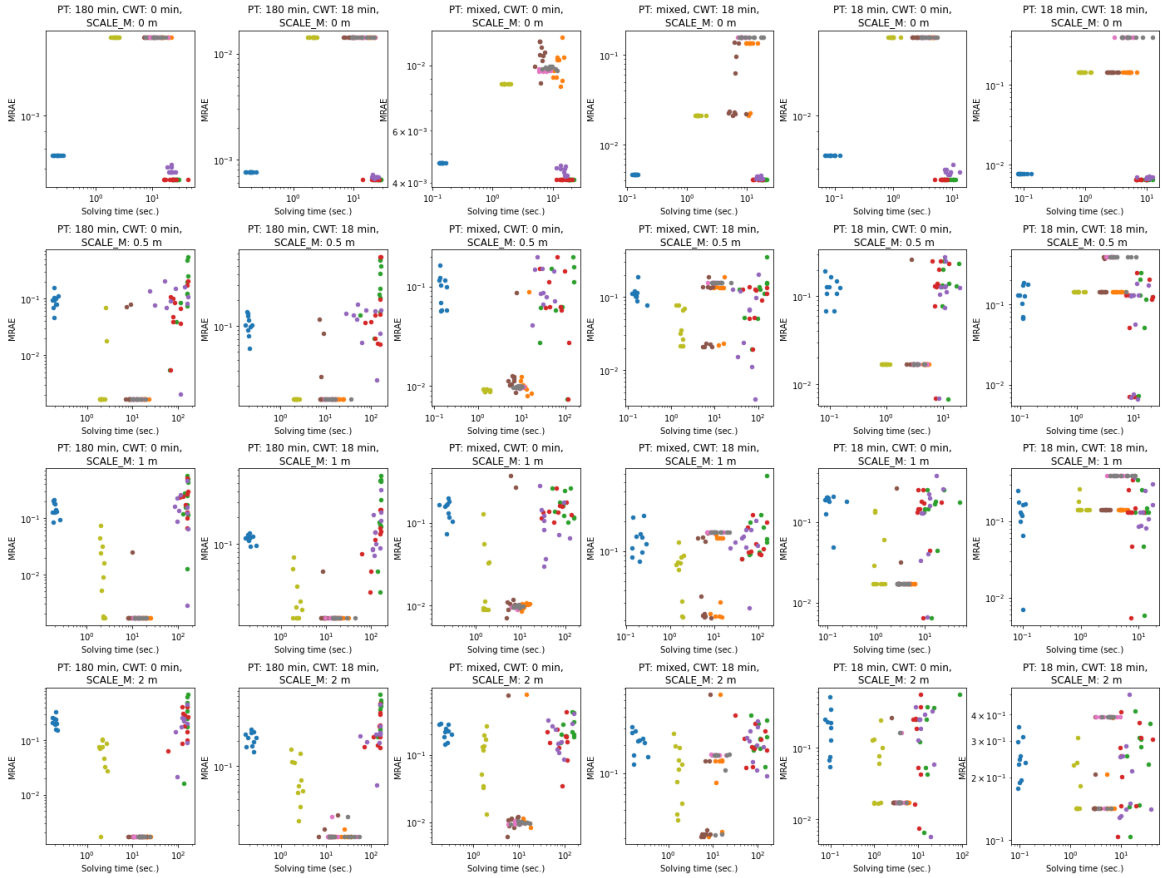

Fig. 2. Mean Relative Absolute Error (MRAE) and solving time of naive methods (● AOI, ● POI), classification-based methods with (● $CP_{TGeoGeo}$, ● CP_{Tnone}, ● CP_{TxGeo}) and without transports (● CP_{GeoGeo}, ● CP_{none}, ● CP_{xGeo}, ● CO_{xPOI}) for different process time distributions (PT), unexpected corridor waiting times (CWT) and measurement error standard deviation ($SCALE_M$).

Acknowledgements

The authors gratefully acknowledge the support of TRUMPF Machine Tools and the Research Training Group GRK 2193 at TU Dortmund University which is funded by the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG)).

References

[1] C. Mieth, P. Humbeck, and G. Herzwurm, "A survey on the potentials of indoor localization systems in production," in *Interdisciplinary Conference on Production, Logistics and Traffic*, pp. 142–154, Springer, 2019.

[2] C. Mieth, "Semantic enrichment of spatio-temporal production data to determine lead times for manufacturing simulation," in *2019 Winter Simulation Conference (WSC)*, pp. 2061–2072, IEEE, 2019.

[3] G. Schroeer, "A real-time uwb multi-channel indoor positioning system for industrial scenarios," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–5, IEEE, 2018.

[4] A. Öztürk, S. Kayalıgil, and N. E. Özdemirel, "Manufacturing lead time estimation using data mining," *European Journal of Operational Research*, vol. 173, no. 2, pp. 683–700, 2006.

[5] G. Ioannou and S. Dimitriou, "Lead time estimation in mrp/erp for make-to-order manufacturing systems," *International Journal of Production Economics*, vol. 139, no. 2, pp. 551–563, 2012.

[6] A. Pfeiffer, D. Gyulai, B. Kádár, and L. Monostori, "Manufacturing lead time estimation with the combination of simulation and statistical learning methods," *Procedia CIRP*, vol. 41, pp. 75–80, 2016.

[7] A. Pfeiffer, D. Gyulai, and L. Monostori, "Improving the accuracy of cycle time estimation for simulation in volatile manufacturing execution environments," *Simulation in Produktion und Logistik 2017*, p. 413, 2017.

[8] M. Mucientes, J. C. Vidal, A. Bugarin, and M. Lama, "Processing times estimation in a manufacturing industry through genetic programming," in *2008 3rd International Workshop on Genetic and Evolving Systems*, pp. 95–100, IEEE, 2008.

[9] H. Reinhardt, M. Münnich, B. Prell, R. Arnold, F. Krippner, M. Weber, F. Seifert, and M. Putz, "Retrieving properties of manufacturing systems from traceability data for performance evaluation and material flow simulation," *Procedia CIRP*, vol. 104, pp. 20–25, 2021.

[10] T. Ruppert and J. Abonyi, "Software sensor for activity-time monitoring and fault detection in production lines," *Sensors*, vol. 18, no. 7, p. 2346, 2018.

[11] P. Charpentier and A. Véjar, "From spatio-temporal data to manufacturing system model," *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 5, pp. 557–565, 2014.

[12] J. Flossdorf, A. Meyer, D. Artjuch, J. Schneider, and C. Jentsch, "Unsupervised movement detection in indoor positioning systems," 2021.

[13] Z. Yang, P. Zhang, and L. Chen, "Rfid-enabled indoor positioning method for a real-time manufacturing execution system using os-elm," *Neurocomputing*, vol. 174, pp. 121–133, 2016.

[14] C. Mieth, A. Meyer, and M. Henke, "Framework for the usage of data from real-time indoor localization systems to derive inputs for manufacturing simulation," *Procedia CIRP*, vol. 81, pp. 868–873, 2019.

[15] S. Gillies *et al.*, "Shapely: manipulation and analysis of geometric objects," 2007–.

[16] L. Perron and V. Furnon, "Or-tools."