



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Ikonen, Teemu; Harjunkoski, Iiro

Incorporation of parameter prediction models of different fidelity into job shop scheduling

Published in: IFAC-PapersOnLine

DOI: 10.1016/j.ifacol.2019.06.051

Published: 01/01/2019

Document Version Publisher's PDF, also known as Version of record

Published under the following license: Unspecified

Please cite the original version:

Ikonen, T., & Harjunkoski, I. (2019). Incorporation of parameter prediction models of different fidelity into job shop scheduling. *IFAC-PapersOnLine*, *52*(1), 142-147. https://doi.org/10.1016/j.ifacol.2019.06.051

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.



Available online at www.sciencedirect.com





IFAC PapersOnLine 52-1 (2019) 142-147

Incorporation of parameter prediction models of different fidelity into job shop scheduling

Teemu J. Ikonen^{*} Iiro Harjunkoski^{*,**}

* Aalto University, School of Chemical Engineering, Kemistintie 1, 02150 Espoo, Finland (e-mail: {teemu.ikonen,iiro.harjunkoski}@aalto.fi). ** ABB Corporate Research, Wallstadter Str. 59, 68526 Ladenburg, Germany

Abstract

Scheduling of industrial job shop processes is normally conducted using estimates of parameters (e.g. processing times) defining the optimization problem. Inaccuracy in these estimated parameters can significantly affect the optimality, or even feasibility, of the scheduling solution. In this work, we incorporate data-driven parameter prediction models of different fidelity into a unit-specific continuous time scheduling model, and investigate the dependency of the solution quality on the prediction model fidelity. Our high-fidelity prediction model is based on Gaussian processes (GP); more specifically we use the maximum a posteriori probability (MAP) estimate. The low and medium-fidelity prediction models rely on determining the average processing time or average processing rate, respectively, from the dataset. In our test case, involving prediction of taxi durations in New York City, the use of GP prediction model yielded, on average, 5.8% and 1.8% shorter realized make spans in comparison to using the low and medium-fidelity prediction models, respectively.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Scheduling Algorithms, Optimization, Parameter Estimation, Machine Learning, Gaussian Processes

1. INTRODUCTION

Industrial job shop processes are inherently stochastic. The optimal scheduling of these processes is dependent on the realization of scheduling parameters (e.g. processing times, customer demand and raw material consumptions). As the information of future realizations is not available, scheduling decisions are typically made based on static tables of estimated scheduling parameters, which contain both inaccuracies and uncertainty.

Modern industrial job shop processes are monitored with a significantly large number of sensors; the number of these sensors being in the order of hundreds or even thousands (Gungor and Hancke, 2009). Useful information on the processes can be extracted from the historical data of these sensors by using machine learning algorithms.

The most common approach of applying machine learning algorithms to job shop scheduling is to learn dispatching rules from historical data. Li and Olafsson (2005) applied an inductive learning algorithm (C4.5) to train a decision tree model using historical data. Later, Olafsson and Li (2010) improved the performance of the model by training it only with a subset of the historical data, representing the best decisions, which were chosen using a genetic algorithm. Mouelhi-Chibani and Pierreval (2010) proposed a neural network-based approach to select, in real time, the best job for a resource, once it becomes available. Instead of using historical data, they trained their model using simulated scheduling data, generated by simulated annealing. For a more extensive review of learning dispatching rules from historical data, the reader may wish to consult the review article by Priore et al. (2014).

Another approach of applying machine learning algorithms to job shop scheduling is to improve the accuracy of the scheduling parameters based on the historical data. Berral et al. (2010) predicted power consumption levels, central processing unit loads and service-level agreement timings of a data center using linear regression and the M5P algorithm. They used these predictions in the scheduling of computing tasks, with an objective of reducing the total power consumption of the data center. Verboven et al. (2008) predicted the runtimes of computing tasks, belonging to the same parameter sweep experiment, using various machine learning tools, in order to assign them to desktop grids with suitable availability windows.

Jiang et al. (2016) applied Gaussian process regression to dynamic scheduling of continuous casting of steel in order to predict slack ratios of jobs. A slack ratio describes the trade-off between the low production time and increased risk of a cast-break. Related to the same application, Zhao et al. (2014) predicted the relationship between adjustable

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved. Peer review under responsibility of International Federation of Automatic Control. 10.1016/j.ifacol.2019.06.051

^{*} Financial support is gratefully acknowledged from the Academy of Finland project "SINGPRO", Decision No. 313466.

gas users and gas tank levels based on a historical dataset using Bayesian Networks. The information was used to ensure safe operating level of the gas tanks.

Regardless of the used estimation model, and the size and quality of the historical dataset, the estimated scheduling parameters contain uncertainty, which is mainly due to stochasticity of the process, measurement errors and model imperfections. The deviations in the realized and estimated scheduling parameters can significantly affect the optimality, or even the feasibility, of the solution.

The purpose of this paper is to accurately predict the processing times from a historical dataset using the Gaussian process (GP) regression, and to use these predictions in job shop scheduling. We benchmark the prediction method against simple average duration and average rate models (see Section 3) by examining the realized make spans of a set of 30 scheduling problems. As the scheduling model, we use the unit-specific continuous-time model by Shaik and Floudas (2009).

2. SCHEDULING PROBLEM

We use the New York City (NYC) taxi duration dataset (Kaggle Inc, 2017)¹, in order to devise a simple scheduling problem that can facilitate the evaluation of parameter prediction models. The dataset contains information of over 1.4 million taxi trips in NYC, including the duration of the trip, passenger count, as well as pick-up and drop-off date, time and coordinates.

We now define a scheduling problem that is based on the dataset but has features that are transferable to industrial job shop scheduling. Let us consider a company, which head quarter is at Wall Street², located at the South West tip of Manhattan. The company performs surveys at remote sites located around NYC. In order to perform a survey, an employee of the company travels to the site by taxi, performs the survey and travels back to the head quarter. We refer to this combination of the outbound taxi trip, the survey and the inbound taxi trip as a *task* (Fig. 1). When performing a survey, an employee relies on a survey team that is based at the head quarter. We consider both taxis and survey teams as processing units, and define the number of both types of units to be two. The number of traveling employees is not restricted. In addition, a taxi that drops off an employee to location A is defined to be instantly available to pick up another employee from location B. The objective of the scheduling optimization is to minimize the make span ms of performing surveys at six different sites located in NYC (see Fig. 2 for an example set of sites). The durations of outbound and inbound taxi trips are predicted (see Section 3), whereas the duration of conducting a survey is fixed to 1800 s.

From an industrial point of view, the taxis and surveying teams are equivalent to machinery or processing units, and the procedure of performing a task to a recipe to produce a product. Alternatively, the inbound and outbound taxi trips can be considered as preparation and







Figure 2. The example scheduling problem, consisting of six survey sites. The grey point indicates the location of Wall Street. Continuous red and dashed blue lines indicate the start and end points of the outbound and inbound trips, respectively, from the test set.

cleaning, respectively, of a processing unit, and the survey as performing the actual job in the unit.

In order to extract the most relevant information from the dataset for our purpose, we create a filtered dataset, in which either the pick-up or drop-off location lies within 500 m from Wall Street (indicated by the red oval in Fig. 3(b)). In addition, we filter trips with unrealistically long durations (>20 000 s). Figure 3(a) shows the histogram of trip durations in the filtered dataset. Further, Figure 3(b) presents a scatter plot of trip durations, in which the location of a point is the pick-up or drop-off location that lies further away from Wall Street, and the color of the point indicates the duration of the trip.

Further, we divide the filtered dataset into training and testing sets, using a ratio of 67-33%. The training set is used to train the prediction model, and the test set to evaluate its accuracy (see the next section). In scheduling, the realization of outbound and inbound trip durations corresponds to the trip in the training set with the closest match to Wall Street and the site coordinates. Figure 2 shows the pick-up and drop-off coordinates (connected by red and blue lines) of these trips for the example set of six sites.

3. DURATION PREDICTION

In the next three sections, we describe our models to predict the duration of a taxi trip, starting from the lowest fidelity. Finally, in Section 3.4, we compare the prediction accuracy of the models.

 $^{^{1}}$ We have amended the dataset with the corresponding fastest routes (in kilometers) from another openly available dataset (Oscarleo, 2017).

 $^{^2~}$ The coordinates of which are 40.70729°N, 74.01095°W.



(a) Histogram of trip durations in the filtered dataset.



(b) Trip durations plotted on the map of New York City. The points of the scatter plot are either the pick-up or drop-off location of the trip, whichever lies further away from Wall Street. The cluster of points in the South East corner corresponds to John F. Kennedy International Airport, and the rectangular area in Manhattan without any points to Central Park.

Figure 3. Visualization of the filtered dataset, containing all taxi trips, the start or end point of which lies within 500 meters from Wall Street. The region is marked by the red circle in Subfigure (b).

3.1 Average prediction model

The simplest approach to estimate the duration of a taxi trip is to calculate the average duration in the dataset and assume that all future taxi trips have the same duration. We refer to the corresponding model as the *average prediction model*. The average taxi duration in the training set is 1082 s. While the model exploits the ground truths of the training set, it naively discards the information stored in the features of the training set. In industrial job shop scheduling, the values of static tables are often determined using the average values of relevant jobs in the past.

3.2 Rate prediction model

Considering the current application, a more reasonable prediction model involves determining the average speed of the taxi trips in the training set, and using the street distance of the planned trip to predict its duration. We refer to this prediction model as the *rate prediction model*. The average taxi speed in the training set is 6.299 m/s. From an industrial point of view, this approach is relevant to estimating the processing time of a job where a given volume of a material state is treated (e.g. heated or purified)³.

3.3 Gaussian process prediction model

A Gaussian process (GP) defines a prior distribution in a space of functions f without parameterizing f. Thus, they are considered as nonparametric methods. After seeing some data, the prior distribution over functions is updated into a posterior distribution over functions. In this work, we use Gaussian process regression ⁴ to predict taxi trip durations. Here, we wish to point out that other nonlinear machine learning methods, such as neural networks or support vector regression, could possibly yield results with similar prediction accuracy.

Let us consider a training set $\mathcal{D} = {\mathbf{x}_i, y_i, i = 1 : N}$, in which the values of \mathbf{y} contain noise in the form of

$$y_i = f(\mathbf{x}_i) + \epsilon, \tag{1}$$

where f is the underlying (noise-free) function, and $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$ is a normally distributed noise term. Using GPs for regression, the posterior predictive density is

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_*|\mu_*, \boldsymbol{\Sigma}_*), \qquad (2)$$

where \mathbf{X} is the design matrix, \mathbf{X}_* is the test input, μ_* is the mean of multivariate distribution and $\boldsymbol{\Sigma}_*$ is the covariance matrix. The latter two are defined as

$$\sum_{\mathbf{k}} = \mathbf{K}_{**} - \mathbf{K}_{*}^{\mathbf{I}} \mathbf{K}_{y}^{-1} \mathbf{K}_{*}, \qquad (4)$$

where, for a chosen kernel $\kappa:$

$$\mathbf{\mathbf{K}}_{*} = \kappa(\mathbf{X}, \mathbf{X}_{*}) \tag{5}$$

$$\mathbf{K}_{**} = \kappa(\mathbf{X}_*, \mathbf{X}_*) \tag{6}$$

$$\mathbf{(} \mathbf{K}_{y} = \kappa(\mathbf{X}, \mathbf{X}) + \sigma_{y}^{2} \mathbf{I}_{N}, \tag{7}$$

where I_N is an identity matrix having dimensions $N \times N$. For a comprehensive review of GPs, the reader may wish to consult text books by Rasmussen and Williams (2006) and Murphy (2012).

We chose to use the exponential kernel

$$\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{||x - x'||}{2l^2}\right),\tag{8}$$

where x and x' are the two inputs for the kernel, σ_f is the scaling parameter, and l is the length scale. The reason is that the kernel enables us to define a priori all taxi durations to be positive.

We include two features from the dataset described in Section 2 in our GP regression model: the latitude and longitude of the pick-up or drop-off location that lies further from the head quarter. Thus, our model predicts the same taxi durations for trips from location A to B and from B to A. In order to enable the relatively memoryintensive training of the model on a laptop, we use a random subset of 3000 samples from the training set.

 $^{^3\,}$ Assuming that the processing time is linearly dependent on the volume.

⁴ The GP regression model is implemented using the Python package PyMC3 (Salvatier et al., 2016).

Parameters σ_f , σ_y and l in Equations 7 and 8 are hyperparameters θ of the model. We tune them using the maximum a posteriori (MAP) estimate

$$\theta_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} \log p(X|\theta) p(\theta),$$
(9)

where $p(\theta)$ is the prior distribution of the hyperparameters (see for example the text book by Murphy (2012)). The tuned hyperparameters θ_{MAP} are listed in Table 1.

Table 1. Hyperparameters of the GP prediction model, estimated using the maximum a posteriori (MAP) method.

parameter	value
σ_{f}	1805
l	0.8151
σ_y	410.7

3.4 Prediction accuracy

We used the three above described models to predict the durations of the taxi trips in the test set. In order to quantify the prediction accuracy, we list the root mean square error (RMSE) and coefficient of determination (r^2) of these predictions in Table 2. As expected, the values of RMSE decreases and r^2 increases with respect to increasing model fidelity. The value of $r^2 \approx 0$ for the average prediction model indicates that the model explains none of the variability around the average duration of the test set. This is also an expected result as the model always returns the average duration of the trips in the training set.

Table 2. Prediction accuracy, measured using root mean square error (RMSE) and coefficient of determination (r^2) , of the average, rate and Gaussian process (GP) prediction models.

prediction model	RMSE	r^2
average	701.8	-2.37e-4
rate	556.5	0.371
GP	438.4	0.610

Figure 4 visualizes the predicted durations by the GP prediction model against actual durations on the cost line drawing of NYC. Predicted durations are shown as a contour plot, and the actual durations of the test set as a scatter plot. In general, the isocurves of the contour are oriented circumferentially around the head quarter. However, the density of the isocurves is the highest in the longitudinal direction of Manhattan, i.e. roughly North East, while they are the sparsest in the direction East from the head quarter.

4. SCHEDULING MODEL

Let us next describe the reasoning for our choice of the scheduling model and how we incorporate the predicted durations from the three alternative prediction models into the scheduling optimization.

As the scheduling model, we use the unit-specific continuous-time model by Shaik and Floudas (2009), which is formulated using the state-task network (STN). The model flexibly facilitates changes in the realized durations, with respect to those predicted. In contrast, the use of discrete time formulations would expose the realized scheduling



Figure 4. Predicted durations (the contour plot) plotted against the actual durations of the test set (scatter points). The predictions are obtained by the GP model.

models to the risk of being infeasible (in the occurrence of a job exceeding its allocated time slot).

The scheduling model by Shaik and Floudas (2009) is an improved version of the original model by Ierapetritou and Floudas (1998) – enabling jobs to span over a number of event points, Δn . In this work, we define our model using equations 1-8, 10-16, 34 and 35 from the paper by Shaik and Floudas (2009), and use the value of $\Delta n = 1$. We determine the total number of event points n iteratively by starting from n = 3, and increasing it by one until the model has a feasible solution. We note that shorter make spans could possibly be obtained by increasing the number of event points, but we have not studied this.

If the durations of inbound and outbound taxi trips would be constant for all tasks, we could formulate the model by using the following four states only: 1) employee in the head quarter (survey not performed), 2) employee on site (survey not performed), 3) employee on site (survey performed), and 4) employee in the head quarter (survey performed). However, in this work, we wish to assign individual predictions of inbound and outbound taxi trip durations to a job. Therefore, we are required to divide these states into substates, specific to a task in the problem. As we have earlier fixed the number of tasks to be six, the number of states in the problem is $4 \times 6 = 24$.

In the current scheduling problem, we only consider wholesale jobs, i.e. the amount of material in each job is required to equal unity. Therefore, we constrain the initially scalar variables that determine the amount of material in a specific job (denoted by symbol b by Shaik and Floudas (2009)) to be equal to the corresponding binary variables defining at which event point the job starts and ends (denoted by symbol w by Shaik and Floudas (2009)).

Finally, after solving the scheduling problem with predicted durations, we evaluate the realization of the scheduling solution. This is conducted by updating the durations of taxi trips with their realized values from the test set, fixing the binary variables of the model, and solving the resulting linear programming problem.

The scheduling model is implemented using the Python package Pyomo (Hart et al., 2011, 2017), and the resulting



Figure 5. Optimized and realized schedules with average, rate and Gaussian process (GP) prediction models. The realized make span (ms) is reported in the captions of Subfigures (b), (d) and (f).

mixed integer linear programming problems are solved using CPLEX.

5. RESULTS

Although the prediction and scheduling models are deterministic, the realized schedules are dependent on individual taxi trips in the test set. Thus, solving the optimization problem with only a single set of sites would not yield representative comparison of the solution quality of the scheduling model with different prediction models. Therefore, we solve 30 different optimization problems⁵, in which the site locations are varied.

In all studied optimization problems, the scheduling model becomes feasible when the number of event points is increased to n = 6. The number is independent of the used prediction model. The resulting optimization problem has 396 binary variables, 601 scalar variables and 4885 constraints.

Let us next examine the optimized and realized schedules (Fig. 5) of the example scheduling problem, the site locations of which were shown in Fig. 2. In both optimized and realized schedules, each task consists of the outbound taxi trip, survey, and inbound taxi trip, in this chronological order.

When using the average prediction model, all taxi trips have the same predicted duration (Fig. 5(a)). In fact, the jobs corresponding to the six tasks could be ordered in any of 6! permutations without changing in the optimized make span. The predicted durations from the rate and GP prediction models, on the other hand, have variation (Figs. 5(c)) and 5(e)). The shortest predicted durations correspond to the sites 1 and 4 lying the closest to the head quarter, whereas the longest predicted duration corresponds to site 2, lying the furthest from the head quarter.

In this case, the realized make span is longer than the optimized make span for all prediction methods. The main cause for this is the significantly longer realized duration for the outbound taxi trip to site 2 in comparison to its predicted values. Other jobs in the realized schedule do not have any outstanding difference in comparison to their estimated values. The realized make span⁶ obtained by the GP prediction model is shorter than those obtained by the average and rate prediction models by the margins of 9.8% and 3.6%, respectively.

Table 3 presents a summary of the results on the 30 optimization problems. We here report the mean and standard deviation of make spans that are normalized with respect to results that would be obtained by an *ideal prediction model*, i.e. a model with the coefficient of determination $r^2 = 1$ and root mean square error of 0, on the same optimization problem. The normalization is conducted by dividing the obtained makespans by the corresponding makespan obtained by the ideal predictor. The normalization is required as the results between the optimization problems are not comparable. The GP

 $^{^5}$ We perform the computations on a laptop, having an Intel $^{\textcircled{R}}$ Core $^{\textmd{TM}}$ i5-7300U processor and 8 GB of memory.

 $^{^{6}}$ It should be noted that comparing the make spans of the optimized schedules, obtained using different prediction methods, is irrelevant as they are solutions to different optimization problems.

prediction model yields, on average, shorter realized make spans than the average and rate prediction models by the margins of 5.8% and 1.8%, respectively. However, these realized make spans are still, on average, 10.9% longer that those that would be obtained by an ideal prediction model.

Our results show that the average computational cost of solving the scheduling problem with predictions from the GP and rate models is an order of magnitude higher than with those obtained from the average model (see Table 3). The reason for this is that, despite the number of variables is the same in all optimization problems, the scheduling problem where all taxi trips have a fixed duration is simpler than with variable durations. As we mentioned earlier, if the taxi trips have a fixed duration, there are at least 6! scheduling solutions (i.e. the permutations of the tasks) that are optimal (in the unrealized domain). In this case, the optimization problem only involves finding the optimal arrangement jobs with respect to each other, whereas with variable durations the problem also involves finding the optimal sequence of tasks.

> Table 3. The summary of results on 30 scheduling problems using the average, rate and Gaussian process (GP) prediction models. For the normalized make span \bar{ms} , we report the mean and the standard deviation (sd).

prediction model	\bar{ms} [-]		solution time [s]		
	mean	sd	mean	\min	max
average	1.178	0.119	4.45	4.25	5.74
rate	1.129	0.072	45.94	7.71	121.69
GP	1.109	0.061	79.66	16.44	344.82

6. CONCLUSIONS

This study studied job shop scheduling with data-driven duration predictions of three levels of fidelity, which we referred to as average, rate and GP prediction models. Our results show that the GP prediction model yields shorter make spans than average and rate prediction models by the margins of 5.8% and 1.8%, respectively.

However, as the scheduling problem becomes more complex with site-specific taxi duration predictions (i.e. the rate or GP model), its computational cost is an order of magnitude higher than when a fixed prediction value is used for all taxi trips (i.e. the average model). These conclusions are specific to the studied scheduling problem.

An advantage of using the GP regression as a prediction model is that it yields, as a by-product, an estimate of the prediction uncertainty. This prediction is a key input parameter for proactive scheduling approaches. The future work will investigate the use of GP regression based parameter predictions in proactive scheduling.

REFERENCES

Berral, J.L., Goiri, Í., Nou, R., Julià, F., Guitart, J., Gavaldà, R., and Torres, J. (2010). Towards energyaware scheduling in data centers using machine learning. In Proceedings of the 1st International Conference on energy-Efficient Computing and Networking, 215–224. ACM.

- Gungor, V.C. and Hancke, G.P. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on industrial electronics*, 56(10), 4258–4265.
- Hart, W.E., Laird, C.D., Watson, J.P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., and Siirola, J.D. (2017). *Pyomo-optimization modeling in python*, volume 67. Springer Science & Business Media, second edition.
- Hart, W.E., Watson, J.P., and Woodruff, D.L. (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3), 219–260.
- Ierapetritou, M.G. and Floudas, C.A. (1998). Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes. *Industrial & engineering chemistry research*, 37(11), 4341–4359.
- Jiang, S.L., Liu, M., Lin, J.H., and Zhong, H.X. (2016). A prediction-based online soft scheduling algorithm for the real-world steelmaking-continuous casting production. *Knowledge-Based Systems*, 111, 159–172.
- Kaggle Inc (2017). New york city taxi trip duration [accessed on the 15th of June, 2018]. URL https:// www.kaggle.com/c/nyc-taxi-trip-duration.
- Li, X. and Olafsson, S. (2005). Discovering dispatching rules using data mining. *Journal of Scheduling*, 8(6), 515–527.
- Mouelhi-Chibani, W. and Pierreval, H. (2010). Training a neural network to select dispatching rules in real time. *Computers & Industrial Engineering*, 58(2), 249–256.
- Murphy, K.P. (2012). Machine learning: a probabilistic perspective. *The MIT Press, Cambridge, MA, USA*.
- Olafsson, S. and Li, X. (2010). Learning effective new single machine dispatching rules from optimal scheduling data. *International Journal of Production Economics*, 128(1), 118–126.
- Oscarleo (2017). New york city taxi with osrm [accessed on the 25th of June, 2018]. URL https://www.kaggle. com/oscarleo/new-york-city-taxi-with-osrm/ version/5.
- Priore, P., Gómez, A., Pino, R., and Rosillo, R. (2014). Dynamic scheduling of manufacturing systems using machine learning: An updated review. AI EDAM, 28(1), 83–97.
- Rasmussen, C.E. and Williams, C.K.I. (2006). Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA*.
- Salvatier, J., Wiecki, T.V., and Fonnesbeck, C. (2016). Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, 2, e55.
- Shaik, M.A. and Floudas, C.A. (2009). Novel unified modeling approach for short-term scheduling. *Industrial* & Engineering Chemistry Research, 48(6), 2947–2964.
- Verboven, S., Hellinckx, P., Arickx, F., and Broeckhove, J. (2008). Runtime prediction based grid scheduling of parameter sweep jobs. In Asia-Pacific Services Computing Conference, 2008, 33–38. IEEE.
- Zhao, J., Wang, W., Sun, K., and Liu, Y. (2014). A bayesian networks structure learning and reasoningbased byproduct gas scheduling in steel industry. *IEEE Transactions on Automation Science and Engineering*, 11(4), 1149–1154.