Simulation in Produktion
und Logistik 2019
Matthias Putz & Andreas Schlegel (Hrsg.)
Wissenschaftliche Scripten, Auerbach 2019

Model-Based Optimisation with Tree-structured Parzen Estimation for Discrete Event Simulation at Container Terminals

Modellbasierte Optimierung mit baumstrukturierter Kerndichteschätzung für ereignisdiskrete Simulation auf Container-Terminals

Marvin Kastner, Nicole Nellen, Carlos Jahn, TU Hamburg, Hamburg (Germany),
marvin.kastner@tuhh.de, nicole.nellen@tuhh.de, carlos.jahn@tuhh.de

Abstract: Traditionally discrete event simulation serves as a tool for estimating the impact of managerial decisions at container terminals. For complex situations, simulation-based optimisation can be employed to lower the amount of executed simulation experiments and therefore reduce the time to obtain results. The search is guided by the already run experiments. This paper examines the applicability of the Tree-structured Parzen Estimator for optimizing simulation models of container terminals. The reasoning is based both on prior literature in the field of simulation and machine learning, as well as a numerical study. Within the experiments, the performance of the Tree-structured Parzen Estimator is compared with Simulated Annealing and Random Search. The Tree-structured Parzen Estimator displays a more explorative and robust search behaviour than Simulated Annealing.

1 Introduction

Seaports are the interface between the various modes of transport in the maritime supply chain. Compared to 2016, the volume of global maritime trade increased by 4 percent in 2017 (UNCTAD, 2018). More container handling and a high system load on the terminals are the result. Therefore, an efficient design of the processes involved is necessary. The handling of containers requires a large number of process steps in the terminal. In addition to loading and unloading the vessels, container transports are also required between the individual functional areas. The assignment of containers to handling equipment is dynamic, with the individual process steps being closely interlinked. This increases the coordination effort of the equipment between the different functional areas and generates stochastic influences that affect the system under consideration. A predictability of the interacting factors is not given.

This paper examines the assignment of containers to horizontal transport vehicles in the container terminal as an example. It can be seen that there is no generally valid optimal allocation of horizontal transport equipment to containers according to a schedule. The process optimisation starts anew for each arriving vessel. A stowage plan determines the location of the export containers on board of the vessels and thus the sequence in which containers are loaded and how they are assigned to quay cranes. In the following, we will use the term quay crane for a ship-to-shore gantry crane. The assignment of import containers to a storage location in the yard is also individual and strongly influences the result. Consequently, the given system complexity and interdependencies between the individual processes are assessed as sufficiently large to justify the use of simulation. Experiments on the real terminal are not possible. Thus, simulation offers a possibility to optimise the terminal processes, whereby different strategies can be compared with justifiable costs and risk expenditure.

In the following, tactical and strategic planning aspects are in focus. During the design phase of a container terminal, simulation is an important tool in decision-making. Based on the infrastructure layouts, number and types of equipment, and the choice of scheduling algorithms, key performance indicators are predicted (Boer and Saanen 2017). The large number of decisions lead to an exponentially growing solution space of possible combinations. For simulation, often a full factorial design is preferred where each parameter combination is tried out (Barton 2010). This is also referred to as grid search in the machine learning community. For reducing the amount of experiments, Barton (2010) suggests a fractional-factorial design in case one intends to examine specific interactions of parameters. For that, the scientist selects a subset of possible parameter combinations, which are of interest for the research question at hand. In case of optimisation, understanding interactions might be a helpful step for finding the optimum. Still, they are not the aim of such a simulation study.

Li et al. (2017) use a multi-fidelity approach to examine each possible parameter combination at a container terminal. The low-fidelity simulation experiments cover the whole parameter grid. They are used to identify the promising parameter configurations for the high-fidelity simulation experiments, which in turn determine the optimum. Al-Salem et al. (2017) suggest that for such large-scale optimisation problems Ordinal Optimisation can be used – a concept initially described by Ho et al. (1992). Here, the best quantile of simulation parameters is guessed based on noisy observations of randomly chosen designs. The probability to be misguided is shown to be low. In industry, a solution that is good enough often fulfils the business objective in case it can be delivered fast and by using few resources. Therefore, obtaining the best quantile of parameters can be sufficient in some cases. Based on the same reasoning, Kotachi et al. (2018) decided to skip certain parameter combinations taking into account the way the system interacts.

In a somehow similar way the machine learning community deals with models which are highly configurable. The models, meaning machine learning algorithms, require a lot of design decisions to start experiments (Pedregosa et al. 2011). This task is so complex that in some cases randomly picking parameter configurations has outperformed human model calibration (Bergstra and Bengio 2012). For supporting the expert in automating the search through a solution space, Hutter et al. (2011) were the first to present an optimisation procedure which can deal with numerical and categorical parameters in a problem-independent way. Eggensperger et al. (2013) compared this approach with two later-emerged approaches, Spearmint and Tree-structured Parzen estimators, for several datasets and showed that the performance of such a hyper-parameter optimisation technique varies with each setup. This fits to the No Free Lunch Theorems in optimisation: One cannot determine the most successful optimisation algorithm for unseen problems (Wolpert and Macready 1997).

2 Simulation-based Optimisation

Zhou et al. (2018, p. 2902) define simulation-based optimisation as a model “to evaluate the performance of the system for a given configuration, while the optimisation algorithm explores alternative configurations in the solution space and identifies the optimal setting”. As the solution space can grow very large and simulations are expensive to run, only a subset is tested in order to find an approximation of the true global optimum. This concept goes by different names, such as “simulation optimisation” (Fu et al. 2005), “simulation-based optimisation” (Zhou et al. 2018), or “simulation evaluation” (Figueira and Almada-Lobo 2014).

Xu et al. (2015) point out that simulation-based optimisation problems have three distinct characteristics: First, the objective value obtained by a simulation run constitutes of a certain proportion of noise with unknown variance. Second, each simulation run turns computationally expensive when the simulation model is more complex. And third, no structural property can be exploited and the simulation model is regarded as a black box. In case there is a known structural property, one might prefer to derive another, simpler representation of the simulation model and optimise the problem with other tools.

One field in which the size of the solution space is too large for a complete evaluation is integrated resource optimisation at container terminals. Kotachi et al. (2018) simultaneously optimise the berth length, the amount of quay cranes, gates, yard trucks, export rows, import rows and the amount of yard cranes per row. Even though not all permissible realistic values are taken into account, the authors calculate 72,576 possible parameter combinations. Kotachi et al. (2018) built an optimisation framework that consists of two stages: First, the interactions between the resources are examined to determine the most promising sequence of resources optimisation tasks. As seven different resources are checked, $7! = 5040$ possible permutations exist. Second, the gained sequence is utilized to optimise each resource one-by-one. The not yet optimised resources are selected according to stochastic sampling. In order to determine the less promising solutions, one needs to make certain assumptions about the black-box property of the simulation model.

Often general guidelines for searching solutions can be used across research domains, so called metaheuristics (Chopard and Tomassini 2018). A generalized process is depicted in Figure 1. Starting with an empty history of model evaluations, first a starting point for the search needs to be determined. Because of lack of information, this might be randomly chosen. The first evaluation of the simulation results is obtained and then recorded in the history. Based on the history the meta-model is updated. This meta-model is a model of the solution space of the examined model (Bergstra et al. 2011). It directs the search and needs to ensure that exploration and exploitation are well balanced. After several iterations a stopping criterion is reached and the best solution so far found is returned as an approximation for the global optimum. This process is also referred to as Sequential Model-based Optimisation (Hutter et al. 2011).While in many publications the term model-based optimisation is used, here the term meta-model is preferred to distinguish between the simulation model to optimise and the probability model of the solution space.

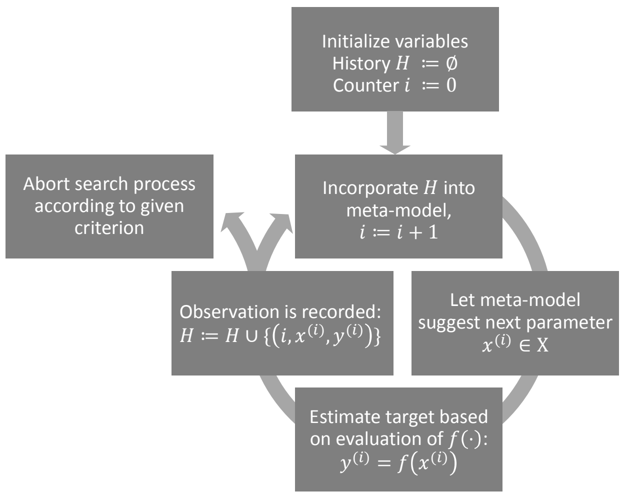
Initialize variables
History $H := \emptyset$
Counter $i := 0$
Incorporate H into meta-model, $i := i + 1$
Abort search process according to given criterion
Observation is recorded: $H := H \cup \{(i, x^{(i)}, y^{(i)})\}$
Let meta-model suggest next parameter $x^{(i)} \in X$
Estimate target based on evaluation of $f(\cdot)$: $y^{(i)} = f(x^{(i)})$

Figure 1: *A sequential meta-model-based optimisation cycle*

Metaheuristics can be applied when the solution space grows too large and/or the evaluation of one possible solution is too time-consuming. In science, computational experiments are a common tool to gain insight into the behaviour of systems that are computable. While discrete event simulation is one of such cases, it is by far not the only one. In different fields of machine learning similar issues arose. Many of the learning algorithms allow detailed configuration of the scientists, which led to a very large solution space. According to Bergstra and Bengio (2012), two common approaches prevailed at the time of their publication: For a manual search, the machine learning expert chooses the next reasonable parameter combination after the current choice of parameters is evaluated. The other option is to explore a predefined grid. The authors showed that random search outperformed these in the sense that after having evaluated a lower amount of configurations, well-performing examples were already obtained. These results might already meet the business objective at hand and no further search needs to be executed (Chapman et al. 2000).

3 Tree-structured Parzen Estimation

The tree-structured Parzen Estimator (TPE) was first introduced by Bergstra et al. (2011) when searching for an appropriate configuration for a Deep Belief Network. Parameters were either categorical variables or continuous variables. Furthermore, dependencies between variables existed: One variable determined the amount of layers of the network and then each layer is configured on its own. Having the configuration of the third layer as part of the solution space is only reasonable if the variable encoding the amount of layers is set to at least three. Therefore, this kind of parameter space is reasonably represented as a tree. This requires specific metaheuristics that support such a tree structure and exploit its properties. The TPE approach has shown good benchmark results (Eggensperger et al. 2013; Bergstra et al. 2013) and the initial paper has been one of the major cited publications in hyper-parameter optimisation in machine learning.

TPE models $p(y < y*)$, $p(x \mid y < y*)$ and $p(x \mid y \geq y*)$, where p denotes a probability density function (short: density), y is a point evaluation of the model, and $x = (x_1, \dots, x_n)$ is a parameter configuration. The first density $p(y < y*) = \gamma$ is a fixed value (e.g. 0.15 was used by Bergstra et al., 2011) set by the experimenter and $y*$ is altered to fit the set value of γ for each iteration. The other two densities can be summarized as $p(x \mid y)$: Given a desired point evaluation value, what are the chances that a certain parameter configuration was used. Commonly TPE is formulated to find a minimum and therefore $p(x \mid y < y*)$ describes the density of parameters which have shown better results whereas $p(x \mid y \geq y*)$ describes which parameters led to a lower performance. As the true probability density functions are unknown, they need to be estimated based on the obtained model evaluations at each iteration.

For each categorical parameter, two probability vectors are maintained and updated: Given the prior vector of N probabilities p_i, the posterior vector elements are proportional to $N \cdot p_i + C_i$ where C_i counts the occurrences of choice i in the so far recorded model evaluations. An example is depicted in Figure 2 (a). The estimator for the better performing models (abbreviated with *better*) and the estimator for the worse performing models (abbreviated with *worse*) are calculated based on the observations already recorded. For each continuous parameter, two adaptive Parzen Estimators are used (Parzen 1962). Given a prior probability density distribution determined by the experimenter, with each point observation obtained from the model, the densities are further approximated. An example is depicted in Figure 2 (b). The parameter choices of the better and worse performing models (abbreviated with *obs.* for observation) are used to create the respective densities (abbreviated with *est.* for estimation). A uniform prior distribution has been assumed.

At each iteration the meta-model consisting of the two densities is used to pick the next parameter configuration x to evaluate. To achieve this, several parameters $\{x\}$ are sampled from the promising distribution $p(x \mid y < y*)$. The parameter configuration x with the highest expected improvement is picked. This criterion is positively correlated with the ratio $p(x \mid y < y*) \,/\, p(x \mid y \geq y*)$ (Bergstra et al. 2011). In Figure 2, this is referred to as *ratio*. The criterion prefers the parameter configuration that has a high probability to lead to small evaluation values and a low probability to obtain too large evaluation values for the minimization problem at hand. After the parameter configuration of the simulation model has been evaluated, the new results are incorporated into the meta-model, i.e. the probability estimators.

4 Simulation Experiment and Result Analysis

Each meta-heuristic needs to prove its applicability to a problem empirically because they exploit different structural properties of the black-box function. Therefore, the TPE is applied to a discrete event simulation model and compared with Simulated Annealing (SA) and Random Search (RS) to get some initial insights in the performance characteristics. The created simulation model is used to assign containers to handling equipment. The focus is on the quay area for waterside handling and the block storage area in the yard. The investigations are limited to a container terminal which is loaded by the arrival and handling of a vessel. Details on the design of the landside traffic connection are not further considered.

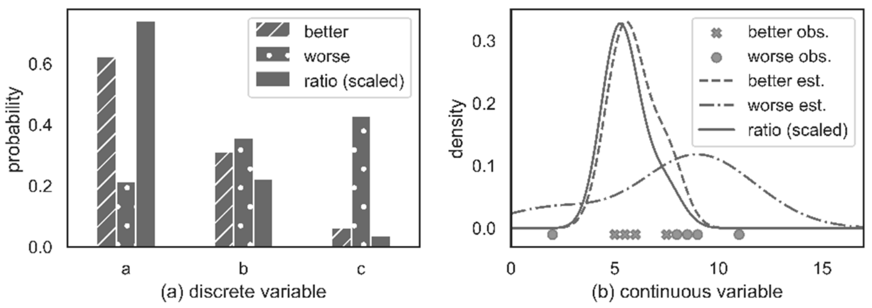

Figure 2: *The expected improvement derived from the two distribution estimations in the discrete and continuous case*

Depending on the experiment, there are up to six quay cranes along the quay for loading and unloading the vessels. It should be noted, that container vessels are not within the system boundary. The quay cranes thus provide the system access for the containers. Yard trucks carry out the horizontal transport between quay and yard. This means that intermediate storage of the containers at the quay is not possible and the quay cranes must transfer the containers directly to the yard trucks. They can equally serve the individual quay cranes and blocks in the yard.

The container storage area in the yard comprises a total of 20 blocks. Storage and retrieval processes in the yard are carried out by RTGs. The travel times of the yard trucks between the functional areas are determined by the route relationships specified by the terminal layout. The containers are transferred to the yard by waiting for the trucks in the yard. The number of yard trucks used is generated at the beginning of the simulation and assigned to a start position in the yard. As long as free jobs are available, they can be taken over by the yard trucks. If a new transport order is successfully determined, the yard truck moves to the location of the container. The waterside unloading process is simulated by creating containers at the respective container gantry crane at the time of their delivery. If no yard truck is available at the quay, the container gantry cranes have to wait. Otherwise, the containers can be transferred directly to the yard trucks. Loaded trucks drive the containers to the designated yard block. There, the truck and container are separated from each other. Process steps for handling export containers can be accepted in the same way as described above, but in reverse order. Although not all unloaded containers are intended for storage in the yard blocks, no attention is paid to transshipment in this work.

The optimisation problem consists of determining the number of quay cranes and trucks need for smooth operation while considering the necessary investments. For this, it is assumed that a quay crane is fifty times more expensive than a truck (Kotachi et al. 2018). Therefore, the loss function is defined to be the product of the inverse simulation time (reflecting the throughput for a fixed number of containers) and the utilization of the equipment weighted by the investment. This leads to the following definition of the loss function:

$$loss = -\frac{1\,day}{t_{simulation}} \cdot \frac{50 \cdot \#QCs \cdot util_{QC} + \#trucks \cdot util_{trucks}}{50 \cdot \#QCs + \#trucks} \tag{1}$$

Here, $t_{simulation}$ is the time used to finish one simulation run (the last container has been successfully moved), $\#QCs$ describes the amount of quay cranes, $\#trucks$ refers to the number of trucks and $util_{equipment}$ describes the ratio the equipment has been busy compared to the overall simulation time. For the optimisation, the amount of quay cranes (3, 4, 5, or 6) serve as a categorical decision variable for different estimators which determine the amount of trucks. For each, a quantized continuous distribution is used (20 to 60 in steps of 1). That leads to 160 possible parameter configurations. All prior distributions are set to uniform. For the implementation of the optimisation, the library *hyperopt* was used (Bergstra et al. 2015).

For the solution space, initially a grid search was executed to explore the response surface. In Figure 3, the weighted utilisation and the loss function are depicted over the number of quay cranes and yard trucks (the grid is in steps of three). It can be seen that the loss function largely differs between the quay cranes and that six quay cranes lead to superior results. Too few yard trucks lead to a low weighted utilization due to waiting quay cranes. Too many yard trucks lead to long waiting times for each of trucks resulting in a low weighted utilization. The minimum detected in the grid was -0.9598 for 6 quay cranes and 30 yard trucks.

For six quay cranes larger amounts of yard trucks don't affect the weighted utilization that much due to the higher investments into the quay cranes already made. The search behavior of SA and TPE is shown in Figure 4 where all picked parameter configurations are visualized. As a baseline method, RS has been employed. Within 30 simulation experiments per optimisation run with one simulation run for each experiment, all metaheuristics detected six quay cranes and approximately 30 yard trucks as the optimal configuration. Due to stochasticity, for 30 optimisation runs SA ranged between 23 and 42, TPE between 29 and 33 and RS between 27 and 43.

More interesting is the slight difference in the search pattern: SA focuses on the better parameters, which can be seen in the high-density values for the number of trucks and the frequency the six quay cranes were chosen. TPE invests more trials in exploring the solution space. Less than six quay cranes are repeatedly tried again. In addition, more often lower and higher numbers of yard trucks are chosen. This has a low effect on finding the optimum, as it can be seen in Figure 5. While RS generally shows a lower performance, SA and TPE arrive at similar optima with TPE showing less outliers in both directions, i.e. better and worse optima. Therefore, TPE seems more robust.

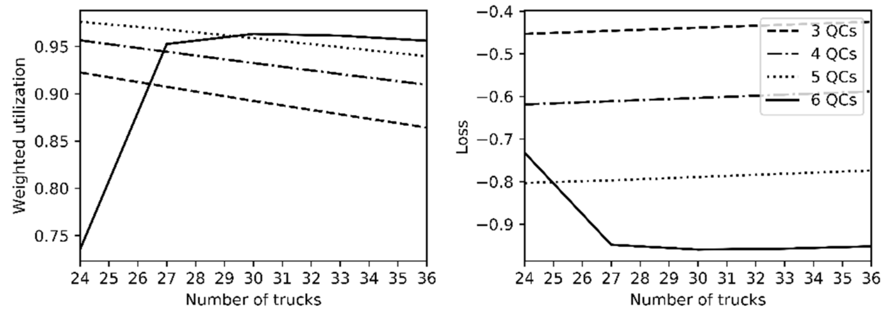

Figure 3: *The weighted utilization and the loss of the simulation model depending on the number of yard trucks and quay cranes*

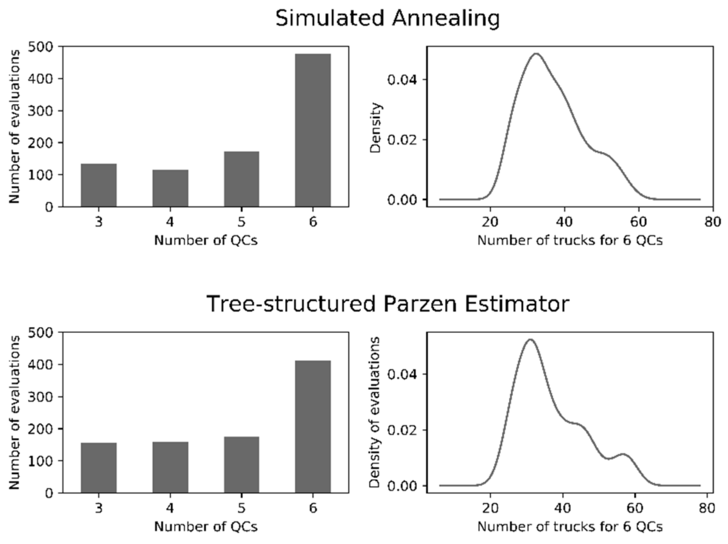

Figure 4: *Comparing the exploration and exploitation behaviour of TPE and SA*

5 Conclusion

In this publication, the authors showed that TPE could be applied to optimise a discrete event simulation model for discrete and continuous parameters. Several numerical experiments were run. With a restricted computational budget, promising parameter configuration ranges were identified which could be further investigated with more simulation runs for each parameter configuration. This empirical sample study is quite limited in several ways. First, due to the No Free Lunch Theorem of Optimisation it is not clear whether empirical results of meta-heuristics can be generalised to other problems (Wolpert and Macready 1997). The applicability of TPE needs to be shown in the future by larger numerical experiments with more repetitions, more variations, and larger solution spaces, possibly examining different optimisation problems at container terminals. This study only showed how the meta-heuristic could be applied to one simulation model designed for one problem. In the future more simulation models representing different container terminals need to be used to proof generalizability. TPE offers many more configuration options which have not been used, e.g. a uniform prior distribution was chosen for both parameters. The search process can be guided by selecting different priors that reflect the belief of the domain expert (Bergstra et al. 2011; Bergstra et al. 2015). Further research needs to be done for how to make use of the domain knowledge of container terminal designers and operators in the best way. The presented approach is meant to be a further source of insights for informed decision-making on or about container terminals. For the machine learning community, Feurer et al. (2015) worked on generalizing found parameter configurations to similar datasets. It remains an open research question whether a similar approach can support the configuration of the TPE algorithm for a more efficient optimisation.

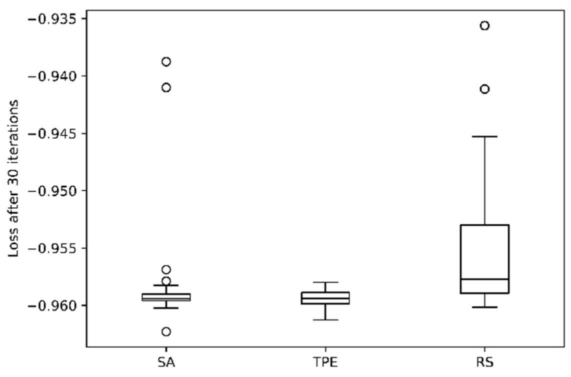

Figure 5: *Comparing thirty executions of SA, TPE and RS with thirty simulation runs each as boxplots*

References

Al-Salem, M.; Almomani, M.; Alrefaei, M.; Diabat, A.: On the optimal computing budget allocation problem for large scale simulation optimization. Simulation Modelling Practice and Theory 71 (2017), pp. 149–159.

Barton, R.R.: Simulation experiment design. In: Johansson, B.; Jain, S.; Montoya-Torres, J.; Hugan, J.; Yücesan, E. (Eds.): Proceedings of the 2010 Winter Simulation Conference (WSC), Baltimore (USA), 5-8 Dec. 2010, pp. 75–86.

Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J.; Zemel, R.S.; Bartlett, P.; Pereira, F.; Weinberger, K.Q. (Eds.): 25th Annual Conference on Neural Information Processing Systems, Granada (Spain), 12-15 December 2011, pp. 2546–2554.

Bergstra, J.; Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (2012) Feb, pp. 281–305.

Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D.D.: Hyperopt: a Python library for model selection and hyperparameter optimization. Computational Science & Discovery 8 (2015) 1, pp. 014008/1-014008/24.

Bergstra, J.; Yamins, D.; Cox, D.D.: Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Dasgupta, S.; McAllester, D. (Eds.): Proceedings of the 30th International Conference on Machine Learning, Atlanta (USA) 16-21 June 2013, pp. 115-123.

Boer, C.A.; Saanen, Y.A.: Using simulation and emulation throughout the life cycle of a container terminal. In: Chan, W.Kin; D'Ambrogio, A.; Zacharewicz, G.; Mustafee, N.; Wainer, G.; Page, E.H. (Eds.): Proceedings of the 2017 Winter Simulation Conference, Las Vegas (USA) 03-06 Dec. 2017, pp. 3126–3137.

Chopard, B.; Tomassini, M.: An introduction to metaheuristics for optimization. Cham: Springer International Publishing 2018.

Eggensperger, K.; Feurer, M.; Hutter, F.; Bergstra, J.; Snoek, J.; Hoos, H.; Leyton-Brown, K.: Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In: NIPS workshop on Bayesian optimization in theory and practice, Lake Tahoe (USA) 10 December 2013, pp. 1–5.

Feurer, M.; Springenberg, J.T.; Hutter, F.: Initializing bayesian hyperparameter optimization via meta-learning. In: Proceedings of the Twenty-Ninth AAAI

Conference on Artificial Intelligence, Austin (USA) 25-30 January 2015, pp. 1128–1135.

Figueira, G.; Almada-Lobo, B.: Hybrid simulation–optimization methods: A taxonomy and discussion. Simulation Modelling Practice and Theory 46 (2014), pp. 118–134.

Fu, M.C.; Glover, F.W.; April, J.: Simulation optimization: a review, new developments, and applications. In: Kuhl, M.; Steiger, N.; Armstrong, F.; Joines, J. (Eds.): Proceedings of the 2005 Winter Simulation Conference, Orlando 4 Dec. 2005, pp. 351–380.

Ho, Y.C.; Sreenivas, R.S.; Vakili, P.: Ordinal optimization of DEDS. In: Ho, Y.C.; Sreenivas, R.S.; Vakili, P.: Discrete Event Dynamic Systems 2 (1992), pp. 61–88.

Hutter, F.; Hoos, H.H.; Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A. (Ed.): Learning and intelligent optimization (2011), pp. 507–523.

Kotachi, M.; Rabadi, G.; Seck, M.; Msakni, M.K.; Al-Salem, M.; Diabat, A.: Sequence-Based Simulation Optimization: An Application to Container Terminals. In: IEEE Technology and Engineering Management Conference (TEMSCON), Evanston (USA) 28 June – 01 July 2018, pp. 1–7.

Li, H.; Zhou, C.; Lee, B.K.; Lee, L.H.; Chew, E.P.; Goh, R.S.: Capacity planning for mega container terminals with multi-objective and multi-fidelity simulation optimization. IISE Transactions 49 (2017) 9, pp. 849–862.

Parzen, E.: On estimation of a probability density function and mode. The Annals of Mathematical Statistics 33 (1962) 3, pp. 1065–1076.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; v. Michel; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; v. Dubourg; Vanderplas, J.; Passos, A.; d. Cournapeau; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12 (2011), pp. 2825–2830.

UNCTAD: Review of Maritime Transport 2018: United Nations 2018.

Wolpert, D.H.; Macready, W.G.: No free lunch theorems for optimization. IEEE transactions on evolutionary computation 1 (1997) 1, pp. 67–82.

Xu, J.; Huang, E.; Chen, C.-H.; Lee, L.H.: Simulation optimization: A review and exploration in the new era of cloud computing and big data. Asia-Pacific Journal of Operational Research 32 (2015) 03, pp. 1–34.

Zhou, C.; Li, H.; Liu, W.; Stephen, A.; Lee, L.H.; Peng Chew, E.: Challenges and opportunities in integration of simulation and optimization in maritime logistics. In: Rabe, M.; Juan, A.A.; Mustafee, N.; Skoogh, A.; Jain, S.; Johansson, B. (Eds.): Proceedings of the 2018 Winter Simulation Conference, Gothenburg (Sweden) 09-12 December 2018, pp. 2897–2908.