DOI: https://doi.org/10.46354/i3m.2019.hms.009

STOCHASTIC FLOATING QUAY CRANE SCHEDULING ON OFFSHORE PLATFORMS: A SIMHEURISTIC APPROACH

D. Souravlias[(a)], M. B. Duinkerken[(b)], S. Morshuis[(c)], D. L. Schott[(d)], R. R. Negenborn[(e)]

[(a), (b), (d), (e)] Delft University of Technology, Delft, The Netherlands
[(c)] Mocean Offshore BV, Amsterdam, The Netherlands

[(a)] d.souravlias@tudelft.nl, [(b)] m.b.duinkerken@tudelft.nl, [(c)] sander@mocean-offshore.com
[(d)] d.l.schott@tudelft.nl, [(e)] r.r.negenborn@tudelft.nl

ABSTRACT

The scheduling of quay cranes is a core logistics challenge that affects significantly the loading and unloading time of a vessel berthed at a container terminal. In this paper, we study the Stochastic Floating Quay Crane Scheduling Problem involving cranes situated on the quay of an offshore modular platform. Specifically, we consider the case in which each crane is situated on a different module of the platform, thereby confining its operation range. Additionally, we assume stochastic crane productivity rates due to the effect of the offshore wind. To tackle the problem, we propose a simheuristic framework, which combines Iterated Local Search with Monte Carlo Sampling into a joint collaborative scheme. The main objective is to minimize the expected completion time of the loading and unloading process taking into account precedence, non-simultaneity, non-crossing, and spatial constraints of the problem at hand. The performance of the proposed simheuristic is investigated on a set of established problem instances across different configuration parameters and under various real-world environmental scenarios offering insightful conclusions.

Keywords: quay crane scheduling, optimization, simulation, offshore platforms.

1. INTRODUCTION

Maritime transport has always been the backbone of international trade and it is expected to maintain its prevailing position in the foreseeable future (Christiansen et al. 2007). Currently, more than 80% of global trade with respect to volume is transported by sea and is handled by seaports. In 2017, world seaborne trade attained a number of 10.7 billion tons of goods and recent predictions have revealed a raise of 3.2% between 2017 and 2022 (UNCTAD 2018). This on-going world transport maritime development has already begun to alter significantly the shape of the contemporary seaports, which require now more than ever sustainable solutions to increase their size. The extension of the port area is not considered as a straightforward task, especially in the case that the land available onshore is fully exploited by the existing facilities. For this reason, an alternative option suggests the expansion of the port towards the sea by adopting new technological developments (Lamas-Pardo et al. 2015).

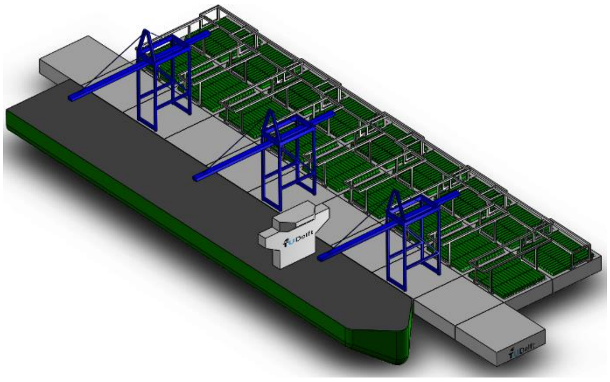

Figure 1: A conceptual design of three quay cranes situated on the quay of the offshore modular platform (Gideonse 2018).

Based on this suggestion, recent technological concepts include the extension of ports via the construction of floating platforms. In this direction, the European project Space@Sea (https://spaceatsea-project.eu/) proposes and investigates the implementation of an offshore modular platform in the proximity of the port of Antwerp. The platform acts as an additional offshore terminal with the major objective of extending the capacity of the port. Under this premise, all operations that occur at a typical terminal may also take place on the floating platform, but with increased complexity due to two main causes. The first cause is that offshore logistics activities are required to respect specific limitations of the platform structure such as the constrained movement of the cranes on the modules of the platform. The second one is that operations on the platform are affected by severe weather phenomena, which are more intense in the open sea. This fact is corroborated by a relevant study which reveals that wind speed offshore is at least 20% higher than onshore (van den Bos 1995).

One of the most essential logistics operations within a terminal that affects significantly the time required for a vessel to stay in the port area is the loading and unloading of its containers. The efficiency of this operation depends critically on the way and the order in which the quay cranes transfer containers, named *crane scheduling*. The so-called Quay Crane Scheduling Problem (QCSP) has received considerable research attention for over than three decades (Daganzo 1989). Up-to-now, a multitude of mathematical formulations along with numerous solution methods have been proposed mainly to tackle deterministic QCSPs (Bierwirth and Meisel 2009, Bierwirth and Meisel 2015, Daganzo 1989, Kim and Park 2004, Legato et al. 2012, Sammarra et al. 2007). In these works, deterministic settings for the involved problem parameters are assumed and, therefore, the uncertainty and dynamics related to container handling operations is not investigated (Al-Dhaheri et al. 2016). Hence, more research studies are needed to further enhance the so-far limited algorithmic artillery on stochastic QCSPs (Al-Dhaheri et al. 2016, Legato et al. 2008).

To the best of our knowledge, in research works that consider stochastic QCSPs, stochasticity is treated as a general concept modelled via arbitrary probability distributions. Even though current research has identified several sources of stochasticity for the QCSP (Chhetri et al. 2016, Tabernacle 1995, Zeng et al. 2011), the impact of any particular uncertainty factor on problem parameters or algorithmic performance still remains unexplored. Besides that, existing studies follow the assumption that the cranes are situated on the quay of onshore terminals, hence the increased complexity and relevant limitations of offshore paradigms are not taken into account. For this reason, the applicability of the proposed QCSP models seems to be questionable in the case that the quay is situated on an offshore floating structure. Additionally, none of the existing studies investigates the impact of environmental conditions on crane productivity rates and in turn how this affects the solution quality of the QCSP.

In this paper, we study the Stochastic Floating Quay Crane Scheduling Problem (SFQCSP) where each crane is situated on a different module of an offshore platform as shown in Figure 1. To perform the loading and unloading operations, the cranes are able to shift in parallel alongside the vessel by using a dedicated rail-track system. Additionally, we make the hypothesis that the platform is equipped with a fully autonomous crane system similar to that used at the Pasir Panjang terminal in the port of Singapore (Gustafsson and Heidenback 2002). Therefore, only unmanned (un)loading operations are performed on the platform. Moreover, we assume stochastic crane productivity rates due to the variability in offshore wind speed. To tackle the problem, we propose a simheuristic framework that combines metaheuristic optimization and simulation into a joint collaborative scheme. Simheuristics have recently emerged as an interesting approach to cope with stochastic combinatorial optimization problems (Juan et al. 2015). Up-to-now, they have been used to provide solutions for several stochastic combinatorial optimization problems (Juan et al. 2014, Michalak and Knowles, 2016).

The proposed simheuristic framework consists of a established local search algorithm, called Iterated Local Search (ILS), which offers high-quality solutions at low computation times (Lourenço et al. 2003). So far, the ILS has been successfully used to tackle combinatorial optimization problems in different application domains (Lourenço et al. 2003). However, to the best of our knowledge, this is the first study where the ILS is applied to the considered problem or any other quay crane scheduling challenge. Additionally, the proposed simheuristic integrates a Monte Carlo Sampling (MCS) approach used to compute the stochastic objective function of the considered stochastic crane scheduling problem (Shapiro 2003). The goal of the developed framework is to minimize the expected (un)loading times of the vessels that berth at the platform under the presence of precedence, non-simultaneity, non-crossing and spatial constraints. Different parameter configurations and various real-world environmental scenarios generated in accordance with the wind state at the location of the platform are used to investigate the performance of the proposed simheuristic.

The remainder of the paper is structured as follows. Section 2 formally defines the considered problem while the employed simheuristic framework is presented in Section 3. The results of the simulation experiments are displayed in Section 4. Conclusions and directions for future research are given in Section 5.

2. STOCHASTIC FLOATING QUAY CRANE SCHEDULING PROBLEM

The studied Stochastic Floating Quay Crane Scheduling Problem (SFQCSP) is considered as a modification of the problem introduced in (Monaco and Sammarra 2011). We assume a set of handling tasks $\Omega = \{1, 2, \dots, n\}$ and a set of quay cranes $Q = \{1, 2, \dots, q\}$. A task stands for loading or discharging a group of containers from the arriving vessel to the offshore floating platform or vice versa. Containers within a group share the same destination and are located at adjacent positions within the same compartment of the vessel, called a bay. Each task $i \in \Omega$ has a specific handling time, p_{i}, and the group of containers of the task are located in a particular bay of the vessel, l_{i}. For the sake of mathematical convenience, but without loss of generality, the beginning and the end of the container service are represented by the dummy tasks 0 and $T = n + 1$, respectively, with $p_0 = p_{\mathrm{T}} = 0$. Based on these tasks, sets $\Omega^0 = \Omega \cup \{0\}$, $\Omega^T = \Omega \cup \{T\}$, and $\bar{\Omega} = \Omega \cup \{0, T\}$, are additionally defined.

Precedence constraints are specified between pairs of tasks located within the same bay. Such constraints are used to determine the ordering of task execution by ensuring that (i) unloading tasks must be performed before loading ones, (ii) loading tasks on the deck must be carried out after those in the hold, and (iii) unloading tasks on the deck must be completed before those in the hold. Also, there are tasks that cannot be processed simultaneously because the containers involved in the tasks are located at adjacent bays which cannot be accessed by quay cranes for safety reasons. Let Φ be the set of task pairs for which a precedence relation exists and let Ψ be the set that includes the task pairs that cannot be processed simultaneously, defined as,

$$\Phi = \{(i,j) | i,j \in \Omega: i \text{ has to be completed before } j\}$$
$$\Psi = \{(i,j) | i,j \in \Omega: i \text{ and } j \text{ cannot be processed simultaneously}\}.$$

We assume that the quay cranes are of the same type sharing identical technical characteristics (e.g. dimension, speed) and can move between adjacent bays within $\hat{t} > 0$ time units. Therefore, the required travel time for a crane from the bay position of task i, l_i, to the bay position of task j, l_j, is computed as $t_{ij} = \hat{t}\,|l_i - l_j|$, whereas the time for crane k from its initial bay position l_0^k to the bay position of task j, l_j is equal to $t_{0j}^k = \hat{t}\,|l_0^k - l_j|$. Moreover, each crane $k \in Q$ adopts an initial bay position l_0^k and a ready time $r^k \geq 0$. All cranes are allowed to preserve a single moving direction (i.e., unidirectional movement), either always handling the task located at the higher-indexed bay or the lower-index bay with respect to their current bay position. Multiple tasks can be processed sequentially by each crane, but each task must be assigned and completed by a single crane, therefore preemption between tasks is not permitted.

As quay cranes are rail mounted on the offshore platform, two crane interference constraints are considered in the SFQCSP. The first one is called the *non-crossing constraint* and prohibits crossing of cranes as they move from one bay to another. The second one is named the *safety constraint* and states that a certain safety distance δ expressed in bay units has to be maintained between any two adjacent cranes. Additionally, a *spatial constraint* is taken into account to reflect the fact that each crane is situated on a different module of the platform. This constraint confines the operation range of a crane, thereby limiting its access to specific groups of bays.

Typically, vessel (un)loading times exhibit different levels of volatility. Stochasticity may originate from various factors such as diverse weather conditions (Chhetri et al. 2016), deviations among operators experience and skill levels (Tabernacle 1995) as well as equipment failure (Zeng et al. 2011). In this study, only unmanned (un)loading operations are performed, hence operators experience and skill is not considered as a factor of uncertainty. Moreover, we do not take into account operation disruptions due to crane breakdowns. Therefore, in this work, weather conditions and specifically the wind speed at the location of the platform is the studied source of uncertainty rending crane productivity rates highly stochastic. To represent the uncertainty in the studied model, we first define crane productivity coefficients, denoted by $\alpha^k \in [0,1]$, $k \in Q$, following the approach in (Legato et al. 2012). The productivity coefficients are determined by sampling crane productivity rates from real-world wind speed/crane productivity data. Then, the handling time of task i assigned to crane k is set to $p_i^k = p_i \backslash \alpha^k$, where p_i is the individual handling time of task i. The objective of the SFQCSP is to determine the expected completion time $E(c_i)$ of each task $i \in \Omega$, such that the expected completion time of the final task T, $E(c_T)$, (i.e., the expected makespan) is minimized. Therefore, the objective function value of the problem solution corresponds to the time required for the entire (un)loading process to complete.

3. SIMHEURISTIC FRAMEWORK

Simheuristics are algorithmic frameworks that combine optimization algorithms and simulation approaches into joint collaborative schemes (Juan et al. 2015). Such frameworks usually employ metaheuristic optimization to tackle the deterministic version of the problem at hand, whereas the computation of the stochastic objective function value is performed through simulation. In this study, the Iterated Local Search (ILS) algorithm (Lourenço et al. 2003) is used as the main optimization method, whereas we employ the Monte Carlo Sampling (MCS) approach (Shapiro 2003) to approximate the value of the expected makespan. Additionally, the proposed simheuristic framework is based on the assumption that high-quality solutions of the deterministic version of a problem are probable to be high-quality solutions also for the stochastic version (Juan et al. 2014). This section presents the employed simheuristic framework by providing details on its optimization and simulation counterparts as well as giving information on their integration.

3.1. Iterated Local Search

ILS is an established metaheuristic algorithm mainly used to tackle combinatorial optimization problems (Lourenço et al. 2003). The algorithm is based on the observation that the iterative application of local search (LS) does not always result in solution improvement. This is because the LS procedure is usually trapped at specific points of the search space, called *local optima*. To mitigate this issue, instead of starting the search from randomly generated solutions, ILS makes use of a specialized mechanism, called *perturbation*. Perturbation involves generating a new solution by applying proper modifications to the incumbent local optimum with the aim of exploring regions beyond the current basin of attraction. In this way, it is possible to probe neighborhoods that the LS heuristic cannot easily reach, thereby amplifying the exploration capabilities of the algorithm.

The algorithm comprises four main components, which are defined prior to its execution. These are a method that generates an initial solution, a perturbation mechanism, a local search heuristic (i.e., neighborhood operator), and an acceptance criterion. The solution generation method creates an initial solution s_0 either randomly or by employing a problem-specific heuristic technique.

Perturbation is applied to the current solution s leading to a modified solution s'. Then, the LS heuristic comes into play and generates a local optimum s'' based on s'. Finally, the acceptance criterion determines which solution will be given as input to the perturbation mechanism in the next cycle of the ILS. The algorithm is executed repeatedly until a predefined termination criterion is satisfied. In this study, the employed termination criterion is the number of local searches applied by the algorithm. Pseudocode of ILS is presented in Algorithm 1. Specifically, the provided pseudocode except for lines 3, 7, and 9 refers to the execution of the employed solution method. For more details on the ILS, the reader is referred to (Lourenço et al. 2003).

In this study, we incorporate the ILS into the proposed simheuristic framework with the aim of determining high-quality solutions for the deterministic version of the problem at hand. To this end, we design the four main components of ILS tailored to the requirements of the considered problem. Next, detailed information is provided on the development of each one of these algorithmic components.

3.1.1. Generation of initial solution

To generate an initial solution for the considered problem, a two-step technique is incorporated into the ILS. In the first step, each crane is assigned the tasks that are within its operation range and cannot be allocated to other cranes. The second step involves distributing the remaining tasks uniformly at random among cranes that are allowed to work on the bays where the containers of the considered tasks are situated. Therefore, the employed generation technique makes certain that the initial solution satisfies the spatial constraints of the SFQCSP. However, it does not guarantee that the solution will respect the other constraints of the problem. To ensure that, the generation procedure is executed repeatedly until the new solution is feasible with respect to all other constraints of the SFQCSP.

3.1.2. Perturbation mechanism

Typically, the design of this mechanism is not a straightforward task as the perturbation has to guide the search away from the current basin of attraction, but not too far leading to a random restart. For the considered problem, the mechanism is developed on the swap of several tasks between each crane and its adjacent ones. The number of tasks swapped between two cranes, called the perturbation step and denoted by p_{st}, plays a significant role in the success of the method. For this reason, its configuration is properly investigated in the experimental part presented in Section 4. The perturbation mechanism is executed repeatedly until it leads to a feasible solution with respect to the constraints of the SFQCSP.

3.1.3. Local search heuristics

For the considered problem, two local search heuristics are developed. The first one, called *shift* heuristic, works on the redistribution of tasks between adjacent cranes.

Algorithm 1 Generic Simheuristic Framework

Input: Problem and algorithm parameters
Output: Solution and its objective function value for the deterministic and the stochastic version of the considered problem

```
1:  $s_0 \leftarrow$ GenerateInitialSolution()
2:  $s \leftarrow$ LocalSearch($s_0$)
3:  $ms \leftarrow$ MonteCarloSampling($s$)
4:  repeat
5:      $s' \leftarrow$ Perturbation($s$)
6:      $s'' \leftarrow$ LocalSearch($s'$)
7:      $ms'' \leftarrow$ MonteCarloSampling($s''$)
8:      $s \leftarrow$ AcceptanceCriterion($s, s''$)
9:      $ms \leftarrow$ BestSolution($ms, ms''$)
10: until termination criterion is satisfied
```

The second one, named *swap* heuristic, interchanges tasks between cranes located in neighboring bays. Assuming an assignment of tasks per crane $\sigma_k,\ k \in Q$ and a unidirectional schedule $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_q)$, the shift heuristic reassigns each task of crane k to cranes $k+1$, $k-1$, located in the upper and lower bays of the current bay, respectively. In the case that any of these cranes is not present, the task is shifted to the existing one. As for the swap heuristic, each task of crane k is assigned to crane $k+1$ and each task belonging to crane $k+1$ is assigned to crane k. For both heuristics, special attention is paid to make sure that each inserted task is placed at the correct position, respecting the lexicographical order within $\sigma_k,\ k \in Q$.

3.1.4. Acceptance criterion

The acceptance criterion determines which solution will be forwarded to the perturbation mechanism at the next cycle of the ILS. Two alternative solutions are compared at each cycle: the local optimum s'' generated by the LS procedure in the current cycle of the algorithm and the local optimum s produced in the previous one. Between these, the solution with the higher quality with respect to the value of the objective function is accepted and used at the next cycle of the algorithm.

3.2. Monte Carlo Sampling

MCS is a computational method that can be used to approximate the objective function value of a stochastic optimization problem (Shapiro 2003). Given an objective function $F(\cdot)$, a probability distribution P and a random sample $\{\omega^1, \omega^2, \dots, \omega^m\}$ of size m drawn from P, a Monte Carlo estimator (also called expected objective function value) of $F(\cdot)$ denoted by $\hat{f}_m(\cdot)$ is defined as:

$$\hat{f}_m(x) = \frac{1}{m}\sum_{i=1}^{m} F(x, \omega^i). \qquad (1)$$

Note that the computation of $\hat{f}_m(\cdot)$ results in a numerical value whose precision depends on m. Typically, samples of larger size (i.e. more observations) lead to more accurate computations of $\hat{f}_m(\cdot)$.

The MCS is employed within the proposed simheuristic framework to compute the value of the expected makespan, which is the used objective function for the SFQCSP. Specifically, the local optimal solution generated by each LS procedure is given as input to the MCS, which is executed until a predefined stopping rule is satisfied. The employed stopping rule is presented in the Section 3.2.1. Lines 3 and 7 of Algorithm 1 refer to the execution of the MCS method. Note that this method is always applied after LS taking into account the hypothesis that there is a relation between high-quality solutions of the deterministic and the stochastic version of the considered problem. Furthermore, a so-called best solution procedure is developed (line 9 of Algorithm 1) to compare the solution for the SFQCSP of the current cycle (ms'') with the solution that was generated in the former cycle (ms). Finally, the solution that achieves the lowest expected makespan is identified as the best solution.

3.2.1. Stopping rule

Determining the exact number of observations needed by MCS to achieve a specific level of precision is considered an intricate task. For this reason, instead of defining a specific sample size, we set a limit on the number of required simulations by computing the relative error of the generated sample. Following the study in (Ata 2007), the relative sampling error is defined as:

$$\text{RSE} = z_{\alpha/2}\sqrt{S_m^2/m}, \quad (2)$$

where $z_{\alpha/2}$ is the z value of the confidence interval at significance level α, m is the number of already performed simulations, and S_m^2 is the variance estimator of the sample. Then, the current simulation is terminated when:

$$\text{RSE} \leq \mu_\varepsilon \bar{X}_m, \quad (3)$$

where $\bar{X}_m$ is an estimator of the sample mean over m observations and $\mu_\varepsilon \in [0,1]$ is the given error threshold, which represents the level of precision. Eq. (3) acts as the stopping rule used to terminate any MCS conducted within the developed simheuristic framework. The benefit from employing this stopping rule is twofold: it enables the simulation to achieve the desirable level of precision while it also averts long and unnecessary simulations, thereby reducing considerably the execution time of the simheuristic.

4. EXPERIMENTAL EVALUATION

The goal of the experimental evaluation is to study the effect of different configurations of the developed approach on the solution quality. To accomplish this, we have conducted extensive simulation experiments adopting the set of instances introduced in (Kim and Park 2004) and also used in (Bierwirth and Meisel 2009, Monaco and Sammarra 2011, Sammarra et al. 2007).

Table 1: Characteristics of the used problem instances including bay ranges per crane (Kim and Park 2004, Monaco and Sammarra 2011).

Set	Instances	$\lvert\Omega\rvert$	$\lvert Q\rvert$	Bays	Bay ranges per crane
A	k13-k22	10	2	1 to 10	[1,7], [3,10]
	k23-k32	15	2	1 to 15	[1,10], [5,15]
B	k33-k42	20	3	1 to 20	[1-10], [5-15], [11-20]
	k43-k49	25	3	1 to 25	[1-12], [8-20], [13-25]

Table 2: Parameter configuration of the developed approach.

Parameter	Description	Value(s)
N_{LS}	Number of applied LS	{51,101}
p_{st}	Perturbation step	{1,2,3}
m_{max}	Max sample size per MCS	2 x 10^4
μ_ε	Error threshold	{1 x 10^{-2}, 5 x 10^{-3}}
α	Significance level	0.95

Characteristics of these instances including limitations imposed on the operation ranges of the quay cranes are shown in Table 1. In this table, note that the number of considered tasks and available cranes are denoted by $|\Omega|$ and $|Q|$, respectively.

Our experimental study comprises two phases. In the first phase, the focus is on determining the best ILS variant with respect to the solution quality for the deterministic version of the problem. In the second phase, we incorporate the best algorithmic variant derived from the previous step into the simheuristic framework. During this step, our aim is to minimize the time required to (un)load a vessel (i.e., expected makespan) under different wind speed/crane productivity scenarios. The parameter configurations used in the experiments are displayed in Table 2.

The simheuristic approach has been developed in Python 3.7.0 using the Anaconda 1.8.7 framework. We performed the experiments on a Windows workstation consisting of an Intel© Xeon 3.70 GHz processor with 32GB of RAM.

4.1. Deterministic case

As described in Section 3.1.3, two specialized neighborhood operators, called shift and swap heuristics, are used by the ILS. Moreover, we consider a low and a high computation budget with respect to the number of applied local searches in order to evaluate the performance of the algorithm under different computation scenarios. Specifically, the low budget case involves carrying out 51 local searches, whereas the high budget case refers to the application of the local search heuristic for 101 times, namely $N_{\text{LS}} \in \{51{,}101\}$. Note that one local search is performed during the initialization phase of the algorithm whereas the remaining local searches are consumed by the main execution of the ILS. Also, we investigate the performance of three perturbation step values, namely $p_{\text{st}} \in \{1{,}2{,}3\}$. Overall, we experiment with 12 different configurations that correspond to an equal number of ILS variants. A number of 10 independent experiments per algorithmic variant and problem instance was conducted

Table 3: Results for the deterministic FQCSP - shift heuristic.

	Low comput. budget			High comput. budget		
Inst.	$p_{\text{st}} = 1$	$p_{\text{st}} = 2$	$p_{\text{st}} = 3$	$p_{\text{st}} = 1$	$p_{\text{st}} = 2$	$p_{\text{st}} = 3$
k13	453.0	453.0	453.0	453.0	453.0	453.0
k14	546.0	546.0	569.4	546.0	546.0	563.1
k15	513.0	513.0	520.2	513.0	513.0	517.8
k16	312.0	326.4	316.8	312.0	319.2	316.8
k17	453.0	453.6	464.1	453.0	453.6	455.7
k18	375.0	379.5	382.2	375.0	380.4	387.6
k19	552.0	556.8	556.8	552.0	562.8	558.0
k20	399.0	399.0	399.0	399.0	399.0	399.0
k21	465.0	465.0	489.0	465.0	481.2	496.8
k22	540.0	540.0	540.0	540.0	540.0	540.0
k23	576.3	576.0	581.1	576.0	576.0	581.4
k24	666.0	666.0	667.8	666.0	666.0	669.6
k25	741.0	742.8	746.4	741.0	741.0	743.7
k26	642.0	642.0	642.3	642.0	642.0	643.2
k27	660.0	660.0	667.5	660.0	660.0	663.9
k28	531.0	531.0	534.9	531.0	531.0	538.8
k29	807.0	813.6	816.0	807.0	807.0	813.0
k30	891.0	895.8	893.4	891.0	900.6	894.6
k31	570.0	570.0	576.0	570.0	570.0	573.0
k32	597.0	597.0	612.3	597.0	597.0	612.3
k33	642.0	642.0	642.0	642.0	649.8	642.0
k34	741.0	741.0	741.0	741.0	741.0	741.0
k35	687.9	686.4	718.2	688.8	684.0	697.8
k36	729.0	729.0	729.0	729.0	729.0	729.0
k37	510.0	510.3	530.7	510.0	510.6	525.3
k38	650.4	651.9	666.3	640.5	648.3	658.2
k39	526.8	525.0	535.5	525.0	525.0	538.2
k40	567.0	567.0	574.8	567.0	567.0	585.0
k41	588.6	589.8	629.7	588.0	589.2	630.9
k42	578.7	591.9	606.3	573.0	573.0	589.2
k43	876.3	899.7	896.4	876.0	888.3	912.9
k44	823.5	835.8	849.9	822.6	830.1	849.0
k45	837.6	840.0	841.2	836.4	839.4	842.4
k46	720.9	723.6	721.8	712.5	717.0	708.0
k47	792.6	793.8	793.2	794.1	793.2	793.2
k48	666.0	670.2	666.0	666.0	666.0	666.0
k49	896.1	902.7	903.3	894.6	898.5	899.4
Mean	625.0	627.7	634.4	**624.2**	626.7	633.2

Table 4: Results for the deterministic FQCSP - swap heuristic.

	Low comput. budget			High comput. budget		
Inst.	$p_{\text{st}} = 1$	$p_{\text{st}} = 2$	$p_{\text{st}} = 3$	$p_{\text{st}} = 1$	$p_{\text{st}} = 2$	$p_{\text{st}} = 3$
k13	453.0	453.0	453.0	453.0	453.0	453.0
k14	546.0	567.0	602.7	546.0	567.0	608.1
k15	531.6	536.4	541.2	543.3	552.6	555.3
k16	312.0	321.6	315.3	312.0	326.4	316.2
k17	453.0	453.9	468.9	453.0	453.6	469.8
k18	375.0	387.6	384.0	375.0	384.9	385.8
k19	552.0	558.0	579.9	552.0	571.8	598.2
k20	399.0	399.0	426.0	399.0	399.0	429.0
k21	465.0	476.7	486.9	465.0	495.6	508.8
k22	599.4	567.0	606.6	568.8	590.4	588.6
k23	576.0	576.0	588.0	576.0	576.0	588.0
k24	670.2	670.2	729.6	669.6	668.4	683.4
k25	741.0	741.0	758.7	741.0	741.0	767.1
k26	642.0	642.0	642.9	642.0	642.0	661.8
k27	660.0	661.2	663.9	660.0	661.8	662.1
k28	531.0	548.7	552.9	531.0	544.8	552.9
k29	807.0	831.3	825.9	807.3	829.8	822.9
k30	891.0	921.6	951.6	891.0	928.2	937.2
k31	570.0	641.7	644.7	570.0	645.0	654.9
k32	597.0	597.3	609.3	597.0	597.3	637.5
k33	642.0	665.4	688.8	642.9	642.0	677.1
k34	756.0	793.5	765.9	756.3	783.0	807.0
k35	690.6	686.7	731.4	688.2	684.0	730.8
k36	729.0	729.0	729.0	729.0	729.0	729.0
k37	510.6	513.9	539.7	510.0	513.0	538.8
k38	647.1	667.5	689.7	640.2	660.0	684.0
k39	528.6	552.9	555.6	528.3	556.8	566.4
k40	567.0	598.5	611.1	567.0	590.7	611.4
k41	593.7	634.2	665.1	588.6	636.3	673.5
k42	580.5	623.7	683.1	573.0	640.2	647.1
k43	936.0	918.6	1056.9	915.0	935.7	957.3
k44	825.9	877.5	832.8	823.2	850.2	864.3
k45	839.4	843.6	842.7	835.2	842.4	837.6
k46	835.5	769.8	854.7	806.7	774.6	807.9
k47	971.7	950.4	983.4	1003.8	918.9	943.8
k48	801.6	747.9	778.5	788.4	740.1	768.6
k49	906.0	927.3	963.3	906.3	930.3	960.6
Mean	641.4	650.0	670.4	**639.3**	650.2	667.2

resulting in 4.440 independent experiments in total. For each experiment, we recorded the best objective function value, which provides an estimate on the time units required to (un)load a vessel that arrives at the offshore platform.

Tables 3 and 4 present the results for the application of the shift and swap heuristics, respectively. To compare the different approaches, we compute the mean value per variant over all considered problem instances. Regarding the shift heuristic, in Table 3 we find that the setting $p_{\text{st}} = 1$ is the best configuration choice for both low and high computation budgets. On the contrary, higher p_{st} values result in worse performance. This leads to the conclusion that mild perturbations should be applied when the considered problem is tackled by the ILS. An additional conclusion is that the high computation budgets exhibit better performance than the low budgets. This result was anticipated since by applying a higher number of local search procedures, usually solutions of better quality are detected. Overall, the best configuration choice combined $p_{\text{st}} = 1$ with the high computation budget scenario achieving a mean objective function value of 624.2.

As for the swap heuristic, in Table 4 we again observe that lower perturbation steps along with higher computation budgets lead to results of higher quality.

This outcome is in line with the best configuration choice that was identified for the shift heuristic. Specifically, we see that using $p_{\text{st}} = 1$ and the high budget choice, the best objective function value for the swap heuristic is attained, which is equal to 639.3. Comparing the shift and swap heuristic, we find superior the performance of the shift heuristic under all considered configurations. This can be attributed to the fact that the swap heuristic works in a similar way to the employed perturbation mechanism. Therefore, there is the risk that the perturbation reverts moves that already applied the swap heuristic, thereby reducing significantly the exploration dynamics of the ILS.

4.2. Stochastic case

In this section, we provide results for the SFQCSP, assuming that the crane productivity rates exhibit uncertainty. Our goal is to minimize the expected time required to (un)load a vessel from/to the offshore platform under different wind conditions. To accomplish this, we use the simheuristic framework that employed the best ILS variant of the previous experimental phase consisting of the shift heuristic and adopting the parameter values $p_{\text{st}} = 1$, $N_{\text{LS}} = 101$. Regarding the simulation component of the framework, we assume a maximum sample size of $\text{m}_{max} = 2 \text{ x } 10^4$ simulations per application of MCS. Additionally, the stopping rule presented in Section 3.2.1 is used to terminate the MCS when the desired level of precision is attained. To this end, two different error threshold values, $\mu_\varepsilon \in \{1 \text{ x } 10^{-2}, 5 \text{ x } 10^{-3}\}$ and a significance level of $\alpha = 0.95$ are considered.

To solve the SFQCSP under realistic wind conditions, we have gathered real-world wind data and computed their impact on crane productivity. Specifically, environmental data from years 1979-2017 containing hourly average wind speeds at a height of 10m at the platform location have been obtained from the DHI MetOcean (http://www.metocean-on-demand.com). In order to provide accurate computations, the wind speed should be measured at the height of 40m (i.e., the crane top level) instead of the height provided in the data. Furthermore, according to (PIANC 2012), the calculations should be based on the 3-second gust speed and not on the typical wind speed. Therefore, the provided hourly average wind speed is converted to a 3-second gust speed at the platform location. Following both suggestions, the required transformation has been performed as follows:

$$U_{40} = U_{10} \text{ x } \left(H_{10}^{40} + F_w^g\right) = U_{10} \text{ x } 1.45, \qquad (4)$$

where U_{10} is the average wind speed at 10m and U_{40} is the 3-second gust speed at 40m. The transformation from 10m to 40m, denoted by H_{10}^{40}, is computed according to the power law relationship $(40/10)^{1/7}$and is equal to 1.22. F_w^g represents the conversion from the average wind speed to the 3-second gust speed and is equal to 1.23 taking into account that the platform is situated more than 20 km away from the coast.

Table 5: Characteristics of the scenarios regarding the wind speed and its impact on crane productivity.

Scenario	Wind speed	3-second gust speed	Crane productivity rate %
1	13.34-14.00	20.00-20.99	[80.00, 100.00)
2	14.01-14.67	21.00-21.97	[60.00, 80.00)
3	14.68-15.33	22.01-22.99	[40.00, 60.00)
4	15.34-16.00	23.00-23.99	[20.00, 40.00)

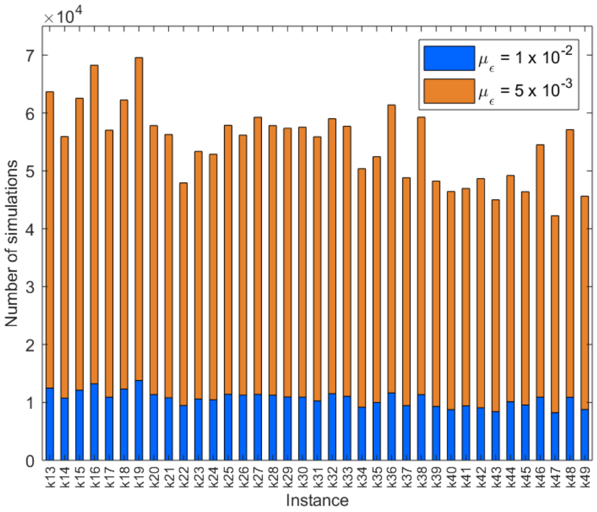

Figure 2: Number of simulations per instance for the used error threshold values for Scenario 1.

Typically, a 3-second gust speed below 20m/s is assumed to enable full crane productivity (100%), while rate values exceeding 25m/s force the crane equipment to cease its operation (PIANC 2012). In this study, we only consider gust speeds between 20 and 24m/s and therefore productivity rates between 20% and 100%. This is because rates equal to 100% imply a deterministic crane scheduling problem while the focus of this section is on stochastic crane scheduling challenges. Additionally, we consider productivity rates less than 20% as an extreme case where crane operations are infeasible, hence obviously the crane scheduling problem is not required to be assessed under these scenarios. Given these assumptions, from the available dataset, we have extracted data items that correspond to gust speeds in [20.00,24.00), resulting in 12612 data points. To compute the crane productivity per data item, we make the hypothesis that productivity rates decrease linearly with respect to the 3-second gust speeds. To study the impact of wind speed variability on crane operations in more detail, we divide the total crane productivity operation range into disjoint groups resulting in four main wind speed/crane productivity scenarios. Information per scenario with respect to the wind speed at 10m, the corresponding 3-second gust speed at 40m along with the corresponding crane productivity rates is shown in Table 5.

Tables 6 and 7 in Appendix A presents the results for the SFQCSP using the four scenarios and the error threshold value $\mu_\varepsilon = 1 \text{ x } 10^{-2}$. Specifically, for each instance, we report the best solution of the deterministic version, c_T, the best solution for the stochastic version of the problem, $E(c_T)$, along with the confidence interval, CI, at significant level $\alpha = 0.95$. Note that there is only one column for the c_T value as the simheuristic detected the same best solution regardless of the employed wind speed/crane productivity scenario. This is because the wind variability does not affect the solutions of the deterministic problem. On the contrary, for each scenario, we see that the higher the wind speed, the lower the crane productivity rate and thus, the less the expected time of the (un)loading process.

Finally, we experiment with the sample size of the MCS for each of the considered error threshold values. Indicatively, Figure 2 reports the total number of simulations performed by the simheuristic per error threshold value regarding Scenario 1. Interestingly, we notice that in order to double the level of precision (i.e., halve the error threshold), the number of simulations has to be increased by about four times. Determining the exact trade-off between precision and execution time depends significantly on the time frame according to which the SFQCSP is required to be solved.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we studied the Stochastic Floating Quay Crane Scheduling Problem in which each crane is situated on one module of an offshore platform. Moreover, we explicitly took into account that the speed of the offshore wind influences the productivity rates of the quay cranes. To address the crane scheduling challenge, we proposed a simheuristic framework that combines the Iterated Local Search algorithm and the Monte Carlo Sampling method into a joint collaborative scheme. The Iterated Local Search algorithm was used to tackle the deterministic version of the problem whereas the Monte Carlo Sampling method was employed to compute the value of the stochastic objective function. Two different local search heuristics, called shift and swap heuristic, were incorporated into the Iterated Local Search algorithm. Experimental results showed the superiority of the shift heuristic under all considered configurations. Additionally, we concluded that only mild perturbations are required when the particular problem is confronted. We have used the developed simheuristic to minimize the time required to (un)load a vessel from/to the platform under different wind speed/crane productivity scenarios. The scenarios were generated by using a specialized approach that quantified the impact of wind speed on crane productivity rates. Future research will involve the application of the simheuristic to larger problem instances including a higher number of cranes and tasks. Additionally, different methods that enhance the efficiency of this framework will be proposed and evaluated on environmental data considering both wind and wave conditions.

ACKNOWLEDGMENTS

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 774253. The opinions in this document reflect only the authors’ view and in no way reflect the European Commission’s opinions. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

Al-Dhaheri N., Jebali A., Diabat A., 2016. A simulation-based Genetic Algorithm approach for the quay crane scheduling under uncertainty. Simulation Modelling Practice and Theory, 66, 122-138.

Ata M.Y., 2007. A convergence criterion for the Monte Carlo estimates. Simulation Modelling Practice and Theory 15 (3), 237-246.

Bierwirth C., Meisel F., 2009. A fast heuristic for quay crane scheduling with interference constraints. Journal of Scheduling 12 (4), 345-360.

Bierwirth C., Meisel F., 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 244 (3), 675-689.

Chhetri P., Jayatilleke G.B., Gekara V.O., Manzoni A., Corbitt B., 2016. Simulating the impact of extreme weather events on port operation. European Journal of Transport & Infrastructure Research 16 (1), 195-213.

Christiansen M., Fagerholt K., Nygreen B., Ronen D., 2007. Maritime Transportation. In: Barnhart C., Laporte G., ed. Transportation, chapter 4, volume 14 of Handbooks in Operations Research and Management Science, Elsevier, 189-284.

Daganzo C.F., 1989. The crane scheduling problem. Transportation Research Part B: Methodological 23 (3), 159-175.

Gideonse P., 2018. Conceptual Harbour Design for the Transport and Logistics Hub of Space@Sea. Technical Report. Delft University of Technology, Delft, The Netherlands.

Gustafsson T., Heidenback C., 2002. Automatic control of unmanned cranes at the Pasir Panjang terminal. Proceedings of the International Conference on Control Applications, volume 1, pp. 180-185, September 18-20, Glasgow, Scotland, UK.

Juan A.A., Barrios B.B., Vallada E., Riera D., Jorba J., 2014. A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. Simulation Modelling Practice and Theory 46, 101-117.

Juan A.A., Faulin J., Grasman S.E., Rabe M., Figueira G., 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives 2, 62-72.

Kim K.H., Park Y.-M., 2004. A crane scheduling method for port container terminals. European Journal of Operational Research 156 (3), 752-768.

Lamas-Pardo M., Iglesias G., Carral L., 2015. A review of Very Large Floating Structures (VLFS) for coastal and offshore uses. Ocean Engineering 109, 677-690.

Legato P., Mazza R.M., Trunfio R., 2008. Simulation-based optimization for the quay crane scheduling problem. Proceedings of the Winter Simulation Conference, pp. 2717-2725, December 7-10, Miami, Florida, USA.

Legato P., Trunfio R., Meisel F., 2012. Modeling and solving rich quay crane scheduling problems. Computers & Operations Research 39 (9), 2063-2078.

Lourenço H.R., Martin O.C., Stützle T., 2003. Iterated local search. In: Glover F.W., Kochenberger G.A., ed. Handbook of Metaheuristics, Springer, 320-353.

Michalak K., Knowles J.D., 2016. Simheuristics for the multiobjective nondeterministic firefighter problem in a time-constrained setting. Proceedings of the European Conference on the Applications of Evolutionary Computation, pp. 248-265, March 30-April 1, Porto, Portugal.

Monaco M.F., Sammarra M., 2011. Quay crane scheduling with time windows, one-way and spatial constraints. International Journal of Shipping and Transport Logistics 3 (4), 454-474.

PIANC Working Group 115, 2012. Criteria for the (un)loading of the container vessels, Technical report.

Sammarra M., Cordeau J.-F., Laporte G., Monaco M.F., 2007. A tabu search heuristic for the quay crane scheduling problem. Journal of Scheduling 10 (4), 327-336.

Shapiro A., 2003. Monte Carlo Sampling Methods. In: Ruszczyński A., Shapiro A., ed. Stochastic Programming, volume 10 of Handbooks in Operations Research and Management Science, Elsevier, 353-425.

Tabernacle J.B., 1995. A study of the changes in performance of quayside container cranes. Maritime Policy & Management 22 (2), 115-124.

UNCTAD, 2018. Review of Maritime Transport 2018, United Nations Publication.

van den Bos W., 1995. Wind influence on container handling, equipment and stacking. Port Technology International Journal, Edition 29.

Zeng Q., Yang Z., Hu X., 2011. Disruption recovery model for berth and quay crane scheduling in container terminals. Engineering Optimization 43 (9), 967-983.

APPENDIX A

In the Appendix, Tables 6 and 7 present detailed results for the SFQCSP under the considered four wind speed/crane productivity scenarios.

Table 6: Results for the SFQCSP under Scenario 1 and Scenario 2.

		Scenario 1		Scenario 2	
Inst.	c_T	$E(c_T)$	CI	$E(c_T)$	CI
k13	453	492.5	[487.6, 497.4]	636.8	[630.5, 643.2]
k14	546	604.1	[598.0, 610.1]	781.9	[774.1, 789.7]
k15	513	566.8	[561.1, 572.4]	733.2	[725.9, 740.5]
k16	312	339.3	[335.9, 342.7]	438.6	[434.2, 443.0]
k17	453	498.2	[493.3, 503.2]	642.7	[636.3, 649.1]
k18	375	413.0	[408.9, 417.1]	535.1	[529.8, 540.5]
k19	552	601.8	[595.8, 607.8]	776.7	[768.9, 784.4]
k20	399	442.4	[438.0, 446.7]	572.1	[566.4, 577.8]
k21	465	520.3	[515.1, 525.5]	673.6	[666.9, 680.3]
k22	540	597.9	[591.9, 603.8]	774.9	[767.2, 782.6]
k23	576	644.6	[638.2, 651.0]	834.0	[825.7, 842.4]
k24	666	744.7	[737.3, 752.1]	964.8	[955.2, 974.5]
k25	741	822.3	[814.1, 830.5]	1067.6	[1056.9, 1078.3]
k26	642	714.7	[707.6, 721.8]	926.3	[917.1, 935.6]
k27	660	733.6	[726.3, 741.0]	949.5	[940.0, 958.9]
k28	531	588.7	[582.8, 594.6]	763.2	[755.6, 770.8]
k29	807	904.7	[895.7, 913.7]	1170.2	[1158.5, 1181.9]
k30	891	1003.2	[993.2, 1013.2]	1303.3	[1290.3, 1316.3]
k31	570	627.7	[621.4, 633.9]	812.5	[804.4, 820.6]
k32	597	661.0	[654.4, 667.6]	858.1	[849.5, 866.7]
k33	642	702.2	[695.2, 709.2]	910.8	[901.7, 919.8]
k34	741	823.1	[814.9, 831.4]	1072.5	[1061.8, 1083.2]
k35	684	777.0	[769.3, 784.8]	1012.7	[1002.6, 1022.8]
k36	729	796.6	[788.6, 804.5]	1034.8	[1024.5, 1045.1]
k37	510	579.0	[573.2, 584.8]	755.3	[747.8, 762.9]
k38	636	716.5	[709.5, 723.6]	927.0	[917.8, 936.3]
k39	525	582.7	[576.9, 588.5]	759.0	[751.4, 766.6]
k40	567	636.8	[630.5, 643.2]	827.1	[818.9, 835.4]
k41	588	671.9	[665.2, 678.6]	873.5	[864.8, 882.2]
k42	573	639.7	[633.3, 646.1]	835.5	[827.1, 843.8]
k43	876	993.3	[983.4, 1003.2]	1298.2	[1285.3, 1311.1]
k44	822	936.5	[927.2, 945.8]	1224.1	[1212.0, 1236.3]
k45	834	954.8	[945.3, 964.3]	1244.5	[1232.1, 1256.9]
k46	705	795.7	[787.8, 803.6]	1033.4	[1023.0, 1043.7]
k47	792	901.9	[892.9, 910.8]	1178.1	[1166.3, 1189.8]
k48	666	731.8	[724.5, 739.0]	948.5	[939.0, 957.9]
k49	894	1016.8	[1006.7, 1026.9]	1321.5	[1308.4, 1334.7]

Table 7: Results for the SFQCSP under Scenario 3 and Scenario 4.

		Scenario 3		Scenario 4	
Inst.	c_T	$E(c_T)$	CI	$E(c_T)$	CI
k13	453	891.7	[900.9, 919.1]	1537.9	[1522.5, 1553.2]
k14	546	1110.5	[744.9, 760.0]	1946.9	[1927.5, 1966.4]
k15	513	1038.3	[1081.2, 1103.1]	1822.8	[1804.5, 1841.0]
k16	312	613.9	[796.9, 813.0]	1066.7	[1056.0, 1077.4]
k17	453	910.0	[947.6, 966.7]	1586.7	[1570.8, 1602.6]
k18	375	752.5	[1081.7, 1103.5]	1318.2	[1305.0, 1331.4]
k19	552	1092.2	[1159.6, 1183.0]	1907.9	[1888.9, 1927.0]
k20	399	805.0	[1350.4, 1377.6]	1414.5	[1400.3, 1428.6]
k21	465	957.2	[1487.1, 1517.1]	1676.1	[1659.4, 1692.9]
k22	540	1092.6	[1295.7, 1321.8]	1919.4	[1900.2, 1938.6]
k23	576	1171.3	[1324.4, 1351.1]	2051.8	[2031.3, 2072.3]
k24	666	1364.0	[1063.4, 1084.8]	2377.4	[2353.6, 2401.1]
k25	741	1502.1	[1636.7, 1669.7]	2631.7	[2605.4, 2658.0]
k26	642	1308.7	[1823.4, 1860.2]	2290.5	[2267.6, 2313.4]
k27	660	1337.7	[1130.4, 1153.2]	2348.7	[2325.2, 2372.2]

k28	531	1074.1	[1196.2, 1220.3]	1863.0	[1844.4, 1881.7]
k29	807	1653.2	[1276.1, 1301.9]	2899.2	[2870.2, 2928.2]
k30	891	1841.8	[1513.2, 1543.7]	3231.0	[3198.7, 3263.3]
k31	570	1141.8	[1434.1, 1463.0]	1991.5	[1971.6, 2011.4]
k32	597	1208.3	[1444.1, 1473.3]	2117.7	[2096.5, 2138.9]
k33	642	1289.0	[1064.8, 1086.3]	2303.1	[2280.1, 2326.1]
k34	741	1528.5	[1300.3, 1326.5]	2735.9	[2708.5, 2763.2]
k35	684	1448.6	[1060.7, 1082.1]	2599.7	[2573.7, 2625.7]
k36	729	1458.7	[1167.0, 1190.5]	2610.7	[2584.6, 2636.8]
k37	510	1075.5	[1233.7, 1258.6]	1926.8	[1907.5, 1946.1]
k38	636	1313.4	[1169.8, 1193.4]	2324.4	[2301.1, 2347.6]
k39	525	1071.4	[1826.9, 1863.7]	1914.2	[1895.0, 1933.3]
k40	567	1178.7	[1720.1, 1754.8]	2119.6	[2098.4, 2140.7]
k41	588	1246.1	[1747.7, 1782.9]	2234.7	[2212.4, 2257.1]
k42	573	1181.6	[1453.4, 1482.8]	2124.5	[2103.2, 2145.7]
k43	876	1845.3	[1660.4, 1693.9]	3321.5	[3288.3, 3354.7]
k44	822	1737.5	[1332.0, 1358.8]	3115.8	[3084.7, 3146.9]
k45	834	1765.3	[1874.2, 1912.0]	3164.8	[3133.2, 3196.4]
k46	705	1468.1	[900.9, 919.1]	2621.2	[2595.0, 2647.4]
k47	792	1677.2	[744.9, 760.0]	3018.2	[2988.0, 3048.4]
k48	666	1345.4	[1081.2, 1103.1]	2400.6	[2376.6, 2424.6]
k49	894	1893.1	[796.9, 813.0]	3384.4	[3350.5, 3418.2]