



# Energy-oriented crane scheduling in a steel coil storage

Christian Weckenborg<sup>1</sup> · Patrick Oetjegerdes<sup>1</sup> · Thomas S. Spengler<sup>1</sup>

Accepted: 28 January 2024 / Published online: 26 February 2024  
© The Author(s) 2024

## Abstract

In steel coil storages, gantry cranes store steel coils in a triangular stacking pattern and retrieve them to serve customer demand on time. The crane movements cause high energy consumption depending on the weight of the steel coils and the direction of the crane movement, which provides a starting point for more efficient crane operation in terms of energy consumption. However, current literature on crane scheduling in steel coil storages and neighboring domains mainly focuses on time-oriented objectives and neglects energy consumption. Therefore, we justify the problem of energy-oriented crane scheduling in steel coil storages and develop a mixed-integer linear programming model and a simulated annealing algorithm. The methods aim to minimize the energy consumed by crane movements while serving customer demand. We present extensive computational experiments comparing the energy-oriented approach against two popular alternatives from the literature. The energy consumption of crane movements can be reduced by 2–22% using energy-oriented crane scheduling compared to the alternatives with an identical customer service level. The simulated annealing algorithm solves instances of the size commonly found in the industrial practice of steel coil storage within an amount of time suitable for practical applications. Since extensive test instances for crane scheduling of steel coil storages have not been available thus far, we make our test instances accessible as a starting point for future research efforts.

**Keywords** Crane sequencing · Place assignment · Warehouse · Test instances · Test set · Benchmark

---

✉ Christian Weckenborg  
c.weckenborg@tu-braunschweig.de

Patrick Oetjegerdes  
p.oetjegerdes@tu-braunschweig.de

Thomas S. Spengler  
t.spengler@tu-braunschweig.de

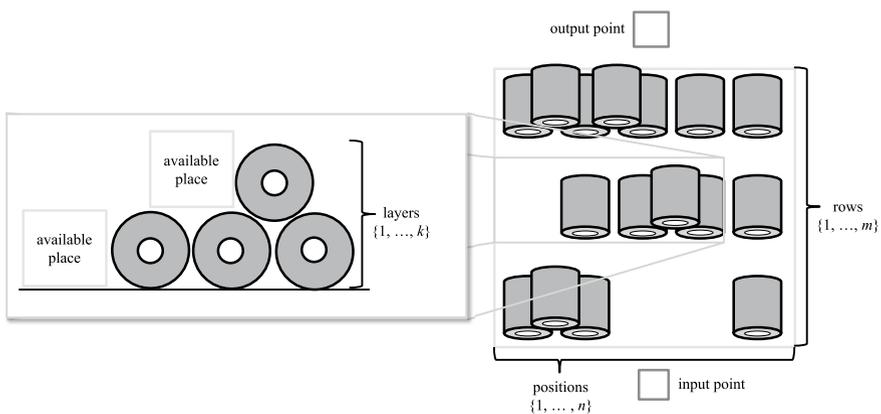
<sup>1</sup> Institute of Automotive Management and Industrial Production, Technische Universität Braunschweig, Mühlentorstr. 23, 38106 Braunschweig, Germany

## 1 Introduction

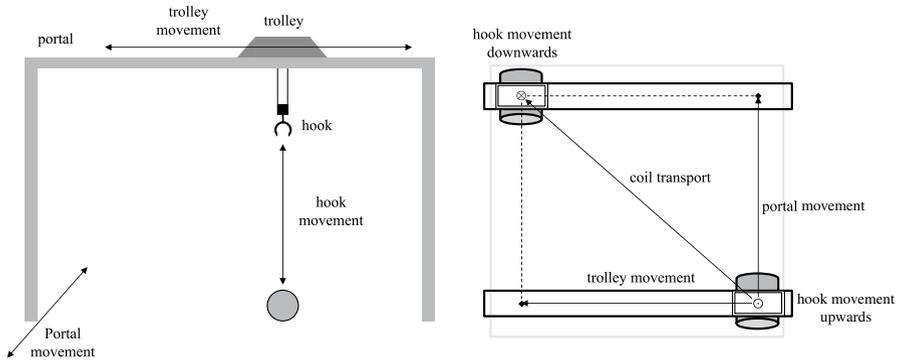
In the manufacturing of flat steel, the refined liquid steel is transformed into solid cuboid steel slabs using continuous casters, converted into hot-rolled steel strips, and coiled up (Puttkammer et al. 2016; Tang et al. 2001; Wichmann and Spengler 2015). The following cold-rolling process changes product-specific attributes like dimension and breaking stresses according to customer specifications. The cold-rolled steel coils are then coated, e.g., with zinc or organic coatings. Consequently, steel manufacturing processes are customer-specific and refer to the casting of individual steel slabs, their reshaping to coils, and surface treatment (Liu et al. 2006).

Along these processes, the steel coils are repeatedly stored in steel coil storages serving as decoupling points in the value chain. Coils entering the storage are called *incoming coils* and require a *storing operation* from the *input point* into the storage area by a specified *deadline*. Coils leaving the storage area are referred to as *outgoing coils* and require a *retrieving operation* to the *output point* of the storage by a specified *deadline*. As coils are stored in multiple *layers* according to a triangular stacking pattern, coils in a lower layer may be *blocked* by coils in an upper layer. Occasionally, *reshuffling* operations may be required to retrieve a blocked coil. The structure of steel coil storages is schematically illustrated in Fig. 1.

Gantry cranes operate the steel coil storage. They consist of the *portal*, *trolley*, and *hook*, allowing for independent movements in three dimensions (cf. Fig. 2). To operate the cranes, electrical energy is consumed depending on the individual movements and coils. As trolley movements require significantly less energy than portal movements, and heavier coils induce a higher energy consumption than lighter coils, the energy consumed by crane movements during a considered period is variable. Considering the energy consumption of cranes in planning their operation might be an excellent lever to reduce the consumed electric



**Fig. 1** Elements of a steel coil storage



**Fig. 2** Elements of a gantry crane and elements of a coil transport

energy of crane movements. The associated planning problem is called *crane scheduling* and covers two interdependent planning tasks. The first planning task involves sequencing the required storing, retrieving, and reshuffling operations and their allocation over time. The second planning task addresses the storage place assignment for the incoming and reshuffled coils. Surprisingly, existing literature on crane scheduling neglects the energy consumption of cranes. In this article, we strive to evaluate the impact of an energy-oriented approach on crane scheduling in steel coil storages regarding potential reduction in energy consumption. To this end, we develop a mixed-integer linear programming approach and a simulated annealing algorithm to simultaneously cover the two interdependent planning tasks associated with energy-oriented crane scheduling in steel coil storages. To derive quantitative evidence on the potential reduction of consumed energy, we conduct extensive computational experiments and compare the results of the energy-oriented approach with two popular alternatives from the literature.

The main contributions of this article are threefold: first, a mixed-integer linear programming approach for energy-oriented crane scheduling in steel coil storages is presented. This is the first exact approach considering the cranes' energy consumption in crane scheduling. Second, the developed simulated annealing algorithm can solve instances typically found in industrial practice, and therefore provides a tool to lift the potential for energy savings in the real-world practice of steel coil storage. Third, we provide our systematically generated test instances as benchmarks for future research efforts.

The remainder of this article is structured as follows: Sect. 2 provides an overview of the associated literature. The general background of energy-oriented crane scheduling and an illustrative example are introduced in Sect. 3. The modeling approach is introduced in Sect. 4, and the simulated annealing algorithm is described in Sect. 5. The computational experiments are conducted in Sect. 6 to show the potential of the proposed approach to reduce energy consumption compared to alternative planning approaches and to illustrate the effectiveness of the developed simulated annealing algorithm. Managerial implications are derived in Sect. 7. The article concludes in Sect. 8.

## 2 Literature overview

General overviews of planning problems in the context of steel manufacturing can be found in Tang et al. (2001) and Xu et al. (2021). This section provides an overview of crane scheduling problems relevant to our approach. These can be found in three different literature streams. First, related literature may be directly associated with crane scheduling in *steel coil storages*. Second, crane scheduling approaches for *steel slab storages* may yield promising results. Third, the yard crane scheduling problem in *container storages* has attracted the attention of researchers.

The crane scheduling problem in *steel coil storages* is first described by Zäpfel and Wasner (2006). They formulate a generalized dynamic job shop model, in which each place in the storage and the crane is modeled as a machine with variable processing times. Compared to a classic job shop problem, they assume processing times of the crane as variable and depending on the routing and sequencing decisions of coils and the dynamic allocation of coils to storage places. Their objective is to minimize the makespan of storing and retrieving operations. Besides a mixed-integer nonlinear program, they develop a local search heuristic to solve the problem and prove that the crane scheduling problem in steel coil storages is NP-hard. Yuan and Tang (2017) propose an integer nonlinear program based on a network flow formulation for the crane scheduling problem. They minimize the makespan of all storing and retrieving operations while limiting the number of reshuffles per coil to one. Further contributions in this stream exclusively consider retrieving operations and neglect storing operations, e.g., for loading trucks. Among these, Tang et al. (2014) formulate a mixed-integer linear program (MILP) to minimize the makespan for retrieving a set of steel coils from a storage area with a single crane and propose a greedy heuristic to solve the problem. Xie et al. (2014) formulate an MILP and develop a greedy heuristic to solve a problem in a similar setting but with two cranes sharing the same rail and a non-interference constraint. Maschitto et al. (2017) formulate an MILP where retrieved steel coils must be loaded on trucks by two overhead cranes operating on the same rail. They aim to minimize the sum of truck loading times. In conclusion, most articles on crane scheduling in steel coil storages are only associated with retrieving operations. Only two articles consider storing and retrieving operations jointly. Additionally, neither of the articles mentioned above considers the energy consumption of crane movements.

*Steel slab storages* are used for decoupling the production processes of casting and hot-rolling, during which the steel slabs cool down and are sorted according to the predetermined sequence required for rolling in the next production stage. Comparable to the problem addressed in the article at hand, a gantry crane stores and retrieves products with a prespecified identity in a storage area. However, the products are not stored according to the triangular stacking pattern. To our knowledge, Rei et al. (2008) first discuss crane scheduling in steel slab storages. They consider one crane and propose a simulation-based optimization approach to evaluate different strategies for place assignment and crane sequencing. They aim to minimize the number of reshuffles while fulfilling all storing and retrieving operations on time. Dohn and Clausen (2010) propose an MILP for scheduling two cranes and develop

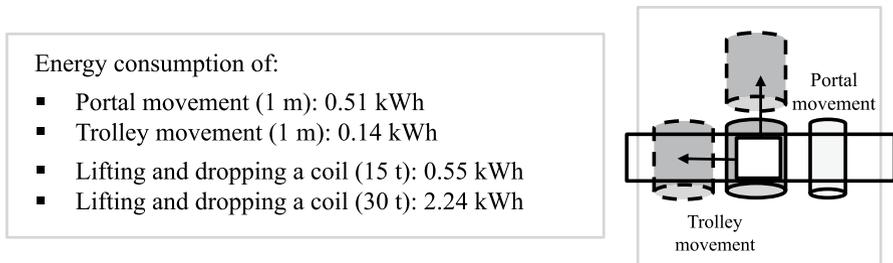
a two-stage greedy heuristic to minimize the maximum tardiness. Zhao et al. (2020) consider cranes that may transport two coils simultaneously. They aim to minimize the makespan of all storing and retrieving operations. Peng et al. (2021) propose an integer linear program (ILP) for storing and retrieving steel slabs with multiple cranes. They solve the problem with a genetic algorithm to minimize the number of reshuffles and the sum of waiting time for all orders. Wang et al. (2021) propose an ILP for scheduling two cranes considering storing and retrieving coils to minimize makespan. In the literature on crane scheduling in steel slab storages, no article considers the energy consumption of crane movements.

The third literature stream addresses yard crane scheduling in *container storages*. Existing research on container handling is vast. An overview of planning problems in container terminal operations is given in Steenken et al. (2004), Stahlbock and Voß (2007), and Gharehgozli et al. (2016). An overview of quay crane scheduling in container terminals is provided by Bierwirth and Meisel (2010) and Bierwirth and Meisel (2015). While the sequencing of crane movements and the storage place assignment are usually considered simultaneously in crane scheduling approaches in steel coil and steel slab storages, the planning tasks are often considered successively or hierarchically in container yards. An overview of scheduling approaches for (yard) cranes without place assignment is provided by Boysen and Stephan (2016) and Boysen et al. (2017), while Lehnfeld and Knust (2014) give an overview of approaches to problems in container terminals focusing on place assignments. To the best of our knowledge, Galle et al. (2018) propose the first simultaneous planning approach. They develop a multi-objective ILP to minimize the crane travel time and the expected number of blocking containers to anticipate the expected number of future reshuffles. Heshmati et al. (2019) consider multiple cranes storing and retrieving containers with release and due dates, minimizing a weighted function of total tardiness and costs for storing products in a specific location. They develop an MILP and a local search heuristic to solve the problem. However, both approaches do not consider the energy consumption of cranes.

To conclude this section, no existing approach has so far considered the energy consumption of crane movements. The next section presents the general background of energy-oriented crane scheduling in a steel coil storage and an illustrative example.

### 3 Energy-oriented crane scheduling in steel coil storage

Steel coil storages decouple adjacent stages of the value chain in terms of time. This is achieved through the storage of coils. The cranes are responsible for storing coils from the preceding stage and retrieving coils for the subsequent stage, considering the requests regarding the timing of storage and retrieval. In terms of time, the portal moves faster than the trolley. However, these movements may co-occur. In terms of energy, an example is provided in Fig. 3, which shows that portal movement consumes more energy compared to trolley movement. A large part of the energy is consumed already at the initial activation of the movement. In contrast, each additional unit of distance moved contributes to a lesser extent to the overall energy consumption.



**Fig. 3** Examples of energy consumption by moving coils in different directions, data based on Labitzke (2011)

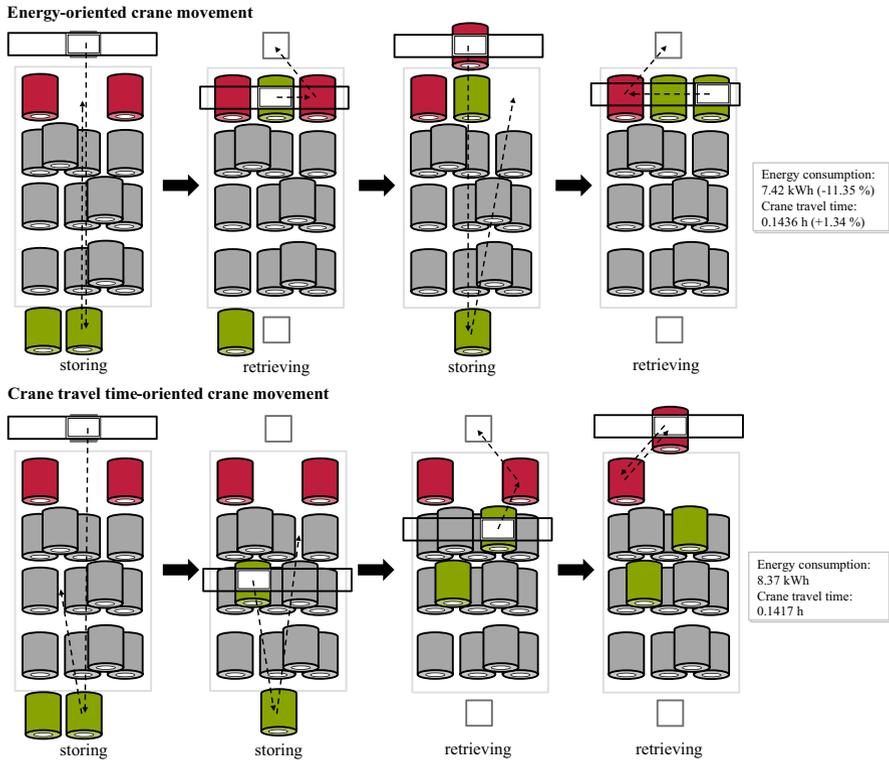
A planning approach that aims for minimal crane travel time may, therefore, prefer a portal movement due to its higher velocity, while an energy-oriented planning approach may favor an exclusive trolley movement due to its lower energy consumption. Additionally, the amount of energy needed to lift and drop the coils strongly depends on the weight of the coil and the height to which it is lifted. Consequently, the overall movements must be carefully coordinated by the crane operator.

The energy-oriented and the crane travel time-oriented planning approaches are illustrated through an example in Fig. 4. In the example, two incoming coils (green) must be stored, and two outgoing coils (red) must be retrieved. In the energy-oriented approach, the first incoming coil is assigned to a place requiring only portal movement (Step 1). Afterwards, the energy-oriented approach chooses a coil to retrieve, which can be accomplished by moving only the trolley due to the assignment of the previous step. After its delivery, the second incoming coil is reached only with a portal movement (Step 2 and Step 3). The last retrieval is again initiated with trolley movement only (Step 4). In contrast, the crane travel time-oriented planning approach frequently uses simultaneous movements of portal and trolley. However, the slight reduction in crane travel time is accompanied by a significant increase in energy consumption.

Certainly, time-oriented planning approaches are suitable for various situations, and they are frequently investigated in scientific literature. A hands-on example is the loading of container vessels, which must be accomplished with a minimal makespan. In the practice of steel coil storages, however, we observe crane utilization rates below 50%. The main challenge in such situations is the storage and retrieval, complying with the requested times to avoid disruptions in the adjacent stages of the value stream. In this context, an energy-oriented planning approach may be favorable. Therefore, in the following section, we develop a model based on mathematical programming for the energy-oriented crane scheduling in steel coil storages.

## 4 Model development

This section presents the model developed for energy-oriented crane scheduling in steel coil storages. To this end, further assumptions of the approach are justified in Sect. 4.1. In Sect. 4.2, the used notation is introduced. The mathematical problem formulation is given in Sect. 4.3.



**Fig. 4** Illustrative example of energy-oriented and crane travel time-oriented crane movements (Color figure online)

## 4.1 Assumptions

Our modeling approach is based on further assumptions regarding the steel coils, the storage area, the gantry crane, the orders, and the discretization of time via stages:

- *Steel coils* may vary in weight. However, they are homogeneous in quality, i.e., each coil can be stacked on top of another.
- The *storage area* comprises two layers. It is homogeneously structured, i.e., each coil can be allocated to each available place. Each place can be identified by its layer, row, and position (cf. Fig. 1). Positions are numbered consecutively, i.e., positions adjacent to one another are in different layers.
- Individual velocities apply for *gantry cranes'* portal, trolley, and hook movements. For each, the velocity is constant. The portal and trolley can move simultaneously with the hook at a prespecified lifting height. During a hook movement, neither the portal nor the trolley may move.
- The neighboring stages of the value chain impose restrictions on the fulfillment of *orders*. Based on the production processes in the upstream stage, the coils

must be stored promptly. Regarding the downstream stage, customer orders must be fulfilled in a timely manner. Therefore, time windows apply for storing and retrieving operations. The time windows must be respected, i.e., tardiness is not acceptable. The crane travel times and the associated energy consumption are known for each coil and combination of places.

- Like Yuan and Tang (2017), we decompose the continuous time of the planning horizon into several discrete *stages*. Each stage allows a maximum of one empty or loaded crane movement and is characterized by its start in continuous time. As the number of stages corresponds to the number of possible crane movements, it must be chosen sufficiently large to avoid limitations of crane movements. Yuan and Tang (2017) use the formula  $2 \cdot (m + n + b)$  to determine the number of stages, where  $m$  is the number of incoming coils,  $n$  the number of outgoing coils and  $b$  the number of blocking coils. However, they assume that coils cannot be placed on top of outgoing coils. This simplification does not apply in our case, meaning we cannot determine the number of blocking coils a priori. We determine the number of stages using the formula  $2 \cdot (m + 3 \cdot n)$ . Consequently, we assume that at most two reshuffle movements, each involving an empty and a loaded move, are required for each outgoing coil. This assumption could prevent finding globally optimal solutions. However, this is a rare scenario. The number of stages is limited globally, not per coil, i.e., if only one reshuffle is needed to retrieve one coil, then three reshuffles are available to retrieve another coil. Thus, for the number of stages to prevent the optimal solution from being found, the optimal solution must require two reshuffles for all but one of the outgoing coils, and three reshuffles for the remaining one. In our case study presented later in this paper, the optimal solution used on average only 52% and at most 59% of the theoretically available stages.

## 4.2 Notation

We adapt and extend the notation published in Yuan and Tang (2017). The storage area comprises storage places  $k, q \in \Psi = \{1, \dots, |\Psi|\}$ . Storage places are allocated in the first ( $\Psi^1 \subseteq \Psi$ ) or the second layer ( $\Psi^2 \subset \Psi, \Psi^2 \cap \Psi^1 = \emptyset$ ) of the storage area. One place of each row is allocated at its first position ( $\Psi^{P_1} \subset \Psi^1$ ) or last position ( $\Psi^{P_{\max}} \subset \Psi^1$ ). Together with the input point  $I$  and output point  $O$  they form the set of all places  $k, q \in \Phi = \{I, O\} \cup \Psi$ .

Coils are comprised in the set  $a \in A = \{1, \dots, |A|\}$ . Coils may already be initially stored or are incoming within the planning horizon  $T$ . Incoming coils  $A_{\text{in}} \subseteq A$  are characterized by their time window for the storing operation  $[\sigma_a^-, \sigma_a^+]$  in continuous time, i.e., pick up from  $I$  must be conducted within the interval. Outgoing coils  $A_{\text{out}} \subseteq A$  are characterized by their time window for the retrieving operation  $[\omega_a^-, \omega_a^+]$ , i.e., dropping at  $O$  must be conducted within this interval. Coils can be part of the incoming and outgoing coils set.

The travel time  $t_{kq}^{\text{load}}$  for the loaded move of coil  $a$  between two places  $(k, q)$  and the associated energy consumption  $E_{kq,a}^{\text{load}}$  are known. Travel time  $t_{kq}^{\text{empty}}$  and energy con-

sumption  $E_{kq}^{\text{empty}}$  of empty moves between two places  $(k, q)$  are known. The large positive number  $M$  serves as a Big-M parameter with  $M > T$ .

The course of time is modeled as continuous. However, discrete movements with a variable duration must be allocated among this continuous time. To this end, discrete stages  $s, \hat{s} \in S = \{1, \dots, |S|\}$  serve to depict a maximum of one empty or loaded move each. The continuous decision variable  $\tau^s \in \mathbb{R}^+$  defines the start time for every stage. Binary decision variables  $W_{kq,a}^s = 1$  indicate that coil  $a$  is moved between places  $(k, q)$  in stage  $s$  (0, otherwise). Binary decision variables  $V_{kq}^s = 1$  indicate an empty move between places  $(k, q)$  in stage  $s$  (0, otherwise). Binary decision variables  $x_{ka}^s = 1$  indicate that coil  $a$  is stored in place  $k$  in stage  $s$  (0, otherwise). An overview of the defined sets, parameters, and variables is given in Table 1.

**Table 1** Definitions of sets, parameters, and variables

Sets and parameters	
$\Psi$	Set of all storage places $k, q \in \Psi = \{1, \dots,  \Psi \}$
$\Psi^1$	Set of all storage places in the first layer $\Psi^1 \subseteq \Psi$
$\Psi^2$	Set of all storage places in the second layer $\Psi^2 \subset \Psi, \Psi^2 \cap \Psi^1 = \emptyset$
$\Psi^{P_1}$	Set of all storage places in the first position $\Psi^{P_1} \subset \Psi^1$
$\Psi^{P_{\max}}$	Set of all storage places in the last position $\Psi^{P_{\max}} \subset \Psi^1$
$I$	Input point of the storage
$O$	Output point of the storage
$\Phi$	Set of all places $k, q \in \Phi = \{I, O\} \cup \Psi$
$A$	Set of all coils $a \in A = \{1, \dots,  A \}$
$A_{\text{in}}$	Set of all incoming coils $A_{\text{in}} \subseteq A$
$A_{\text{out}}$	Set of all outgoing coils $A_{\text{out}} \subseteq A$
$S$	Set of all stages $s, \hat{s} \in S = \{1, \dots,  S \}$
$t_{kq}^{\text{empty}}$	Travel time of the empty crane movement between places $k, q \in \Phi$
$t_{kq}^{\text{load}}$	Travel time of the loaded crane movement between places $k, q \in \Phi$
$E_{kq}^{\text{empty}}$	Energy consumption of the empty crane movement between places $k, q \in \Phi$
$E_{kq,a}^{\text{load}}$	Energy consumption of the loaded crane movement with coil $a \in A$ between places $k, q \in \Phi$
$\sigma_a^-$	Release date of storing order of coil $a \in A$
$\sigma_a^+$	Deadline of storing order of coil $a \in A$
$\omega_a^-$	Release date of retrieval order of coil $a \in A$
$\omega_a^+$	Deadline of retrieval order of coil $a \in A$
$T$	Planning horizon
$M$	Big-M parameter $M > T$
Decision variables	
$V_{kq}^s$	Binary variable with value 1, if an empty movement between places $k, q \in \Phi$ in stage $s \in S$ happens, 0 otherwise
$W_{kq,a}^s$	Binary variable with value 1, if a loaded movement with coil $a \in A$ between places $k, q \in \Phi$ in stage $s \in S$ happens, 0 otherwise
$x_{ka}^s$	Binary variable with value 1, if coil $a \in A$ occupies place $k \in \Phi$ at stage $s \in S$ , 0 otherwise
$\tau^s$	Non-negative continuous variable indicating the starting time of stage $s \in S$

### 4.3 Mathematical model

The objective function (1) minimizes the energy consumption of all loaded and empty crane movements.

$$\min E = \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{s \in S} \sum_{a \in A} (E_{kq,a}^{\text{load}} \cdot W_{kq,a}^s) + \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{s \in S} (E_{kq}^{\text{empty}} \cdot V_{kq}^s) \tag{1}$$

$$\sum_{q \in \Phi | q \neq I} \sum_{s \in S} W_{Iq,a}^s = 1 \quad \forall a \in A_{\text{in}} \tag{2}$$

$$\sum_{k \in \Phi | k \neq O} \sum_{s \in S} W_{kO,a}^s = 1 \quad \forall a \in A_{\text{out}} \tag{3}$$

$$\sum_{k \in \Phi | k \neq O} \sum_{s \in S} W_{kO,a}^s = 0 \quad \forall a \in A \setminus A_{\text{out}} \tag{4}$$

$$\tau^s - \sigma_a^+ \leq (1 - x_{Ia}^s) \cdot M \quad \forall a \in A_{\text{in}}, s \in S \tag{5}$$

$$\sigma_a^- - \tau^s \leq x_{Ia}^s \cdot M \quad \forall a \in A_{\text{in}}, s \in S \tag{6}$$

$$\tau^s + \sum_{k \in \Phi | k \neq O} (W_{kO,a}^s \cdot t_{kO}^{\text{load}}) - \omega_a^+ \leq x_{Oa}^s \cdot M \quad \forall a \in A_{\text{out}}, s \in S \tag{7}$$

$$\omega_a^- - \left( \tau^s + \sum_{k \in \Phi | k \neq O} (W_{kO,a}^s \cdot t_{kO}^{\text{load}}) \right) \leq (1 - x_{Oa}^s) \cdot M \quad \forall a \in A_{\text{out}}, s \in S \tag{8}$$

Equations (2)–(8) ensure that all orders are fulfilled and the time windows for the storing and retrieving operation are respected. Equation (2) ensures that all incoming coils are placed in the storage area or are directly transported to the output point, while all outgoing coils have to be placed on the output point according to (3). Only outgoing coils can be placed on the output point, according to (4). Equation (5) ensures that a coil is not allowed on the input point after its storing deadline  $\sigma_a^+$ . According to (6), a coil must remain on the input point until its storing release date  $\sigma_a^-$ . Equation (7) describes that a coil must arrive at the output point before its retrieval deadline  $\omega_a^+$ . Equation (8) prevents a coil to be placed on the output point before its retrieval release date  $\omega_a^-$ . Please note that these constraints ensure the correct sequence of storing and retrieving for each coil also in cases of overlapping storing and retrieving time windows.

$$\tau^s \geq \tau^{s-1} + \sum_{k \in \Phi} \sum_{q \in \Phi} (t_{kq}^{\text{empty}} \cdot V_{kq}^{s-1}) + \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{a \in A} (t_{kq}^{\text{load}} \cdot W_{kq,a}^{s-1}) \quad \forall s \in S |_{s \neq 1} \tag{9}$$

$$\tau^1 = 0 \tag{10}$$

Equations (9)–(10) define the structure of time. Equation (9) enforces the starting time of any but the initial stage to depend on the starting time of the previous stage and the duration of the activity of the previous stage. The formulation as an inequality allows waiting times of the crane within the stages. Equation (10) defines the starting time of the initial stage as zero.

$$\sum_{k \in \Phi} x_{ka}^s = 1 \quad \forall a \in A, s \in S \tag{11}$$

$$\sum_{a \in A} x_{ka}^s \leq 1 \quad \forall a \in A, s \in S \tag{12}$$

$$\sum_{a \in A} \sum_{k \in \Phi} W_{kl,a}^s = 0 \quad \forall s \in S \tag{13}$$

$$\sum_{a \in A} \sum_{k \in \Phi} W_{Ok,a}^s = 0 \quad \forall s \in S \tag{14}$$

Equation (11) ensures that every coil  $a$  is located at exactly one place  $k$  at every stage. As defined by (12), there may be upmost one coil stored at each storage place per stage. Equation (13) prohibits loaded moves to the input point and (14) prohibits loaded moves from the output point.

$$\sum_{k \in \Phi} \sum_{q \in \Phi} V_{kq}^s + \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{a \in A} W_{kq,a}^s \leq 1 \quad \forall s \in S \tag{15}$$

$$\sum_{q \in \Phi} \sum_{a \in A} W_{kq,a}^s = \sum_{q \in \Phi} V_{qk}^{s-1} \quad \forall k \in \Phi, s \in S|_{s \neq 1} \tag{16}$$

$$\sum_{q \in \Phi} V_{kq}^s - \sum_{q \in \Phi} \sum_{a \in A} W_{qk,a}^{s-1} \leq 0 \quad \forall k \in \Phi, s \in S|_{s \neq 1} \tag{17}$$

$$x_{ka}^s = x_{ka}^1 - \sum_{q \in \Phi} \sum_{\hat{s}=1}^{s-1} W_{kq,a}^{\hat{s}} + \sum_{q \in \Phi} \sum_{\hat{s}=1}^{s-1} W_{qk,a}^{\hat{s}} \quad \forall s \in S|_{s \neq 1}, k \in \Phi, a \in A \tag{18}$$

According to (15), only one loaded or empty move is allowed in each stage. Equation (16) ensures that loaded moves start where the previous empty move ended. Analogously, (17) ensures that an empty move starts where the previous loaded move ended. Please note that the valid inequality allows ending movements in the planning horizon after a loaded move if all orders have been fulfilled. Equation (18) maps the storage occupation of coils to the movements conducted with them, taking the initial location of the coil into account.

$$\sum_{q \in \Phi} \sum_{a \in A} W_{kq,a}^s \leq 1 - \sum_{a \in A} x_{k+1,a}^s \quad \forall s \in S, k \in \Psi^1 \setminus \Psi^{P_{\max}} \quad (19)$$

$$\sum_{q \in \Phi} \sum_{a \in A} W_{kq,a}^s \leq 1 - \sum_{a \in A} x_{k-1,a}^s \quad \forall s \in S, k \in \Psi^1 \setminus \Psi^{P_1} \quad (20)$$

$$2 \cdot \sum_{k \in \Phi} \sum_{a \in A} W_{kq,a}^s \leq \sum_{a \in A} (x_{q-1,a}^s + x_{q+1,a}^s) \quad \forall s \in S, q \in \Psi^2 \quad (21)$$

Equations (19)–(21) refer to the triangular stacking pattern in steel coil storages. Loaded moves from a storage place in the lower layer are allowed only if the neighboring storage place with a higher position index is not occupied, according to (19). Analogously, (20) enforces this for the neighboring storage place with a lower position index. Therefore, the occasional blocking of coils in the lower layer is considered. Please note that these constraints do not apply to the first or last positions, respectively. Equation (21) defines that a loaded movement can only end at a storage place in the second layer if the two neighboring storage places are occupied.

$$\sum_{k \in \Phi} V_{k_0^c k}^1 = 1 \quad (22)$$

$$W_{kq,a}^s \in \{0, 1\} \quad \forall k \in \Phi, q \in \Phi, s \in S, a \in A \quad (23)$$

$$V_{kq}^s \in \{0, 1\} \quad \forall k \in \Phi, q \in \Phi, s \in S \quad (24)$$

$$x_{ka}^s \in \{0, 1\} \quad \forall k \in \Phi, s \in S, a \in A \quad (25)$$

$$0 \leq \tau^s \leq T \quad \forall s \in S \quad (26)$$

Equation (22) initializes the model with an empty move from the initial position of the crane  $k_0^c$  to any place. Equations (23)–(26) define the range of decision variables.

## 5 Development of a simulated annealing algorithm

The crane scheduling problem in steel coil storages is a short-term operational planning problem and requires methods that identify sufficiently good solutions within an appropriately short time. To this end, we develop a simulated annealing algorithm consisting of a construction heuristic to generate initial solutions, and an improvement procedure. In this section, we first introduce the concept of simulated annealing, followed by the introduction of the construction heuristic. Subsequently, the different problem-specific operators to find neighboring solutions by modifying previous solutions are discussed. Finally, the parameters of the simulated annealing algorithm are introduced.

Simulated Annealing (SA) is a metaheuristic framework inspired by the cooling behavior of liquids such as metals or glass, during which the substance crystallizes in an energy-minimal configuration of the atoms in the solid state. This behavior is adopted by the SA framework introduced by Kirkpatrick et al. (1983). The procedure starts with an initial solution and subsequently explores the solution space through neighboring solutions. To avoid getting stuck in local optima, worse neighboring solutions are accepted with a certain probability depending on the so-called temperature. Following the decrease in temperature, the probability of accepting worse solutions reduces. The procedure intends to converge toward an optimal solution.

Our algorithm is described in the following. First, a construction heuristic is executed to generate an initial solution. As our problem is similar to the pickup and delivery problem with time windows, we adapt an insertion-based construction heuristic from the literature (Lu and Dessouky 2006). The general idea is to initially construct a valid crane tour only consisting of retrieval orders, as their start and end points are known a priori. Subsequently, the other orders are iteratively inserted into this sequence. In our algorithm, the retrieval orders are sequenced with an increasing release date of the retrieval order. As one or two other coils may block a coil from being retrieved, reshuffling may be necessary before the retrieval. The reshuffles do not have to be performed immediately before the retrieval but at any earlier point in the sequence. However, a place assignment is required for the blocking coils. Therefore, for each possible position in the sequence where the reshuffling can be conducted, the place with the lowest increase in energy consumption of the current sequence is determined for the reshuffled coil. Subsequently, the reshuffling operation is allocated to the position in the sequence, which results in the lowest increase in energy consumption. Inserting the reshuffling operations into an existing sequence may change the starting positions of already planned movements. The correct consideration of movements is ensured through updating the place assignments. Further, it is ensured that the time windows for each order are met, and orders are moved to an earlier position in the sequence if necessary. Finally, the storing orders are inserted similarly.

The constructed initial solution is subsequently altered during the improvement procedure. To this end, two operators for adjusting the existing solution are considered. One operator focuses on altering the sequence of crane movements, the second operator alters the place assignment. The sequence can be altered if the time windows of the associated orders overlap. Therefore, this operator refers to a sequence shift of two consecutive orders (cf. Fig. 5a, orders R2 and S3). The second operator alters the place assignment. To this end, a storing or reshuffling order in the sequence is selected randomly, and a different available place is randomly designated (cf. Fig. 5b, green coil). After both operators, the place assignment of all succeeding storing and reshuffling orders is updated according to the procedure described in the construction heuristic (cf. Fig. 5b, red coil).

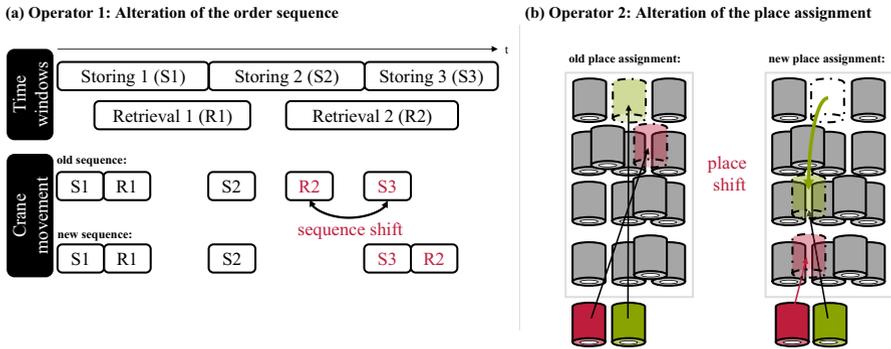


Fig. 5 Operators applied in the improvement procedure (Color figure online)

Two additional functions control the algorithm, i.e., the cooling function and the acceptance function. The cooling function describes the temperature behavior over the iterations of the algorithm. We use the cooling function of Lundy and Mees (1986) which has shown promising results in the field of crane scheduling problems (Lee et al. 2007). Accordingly,  $T_{k+1} = \frac{T_k}{1+\beta \cdot T_k}$ , where  $\beta$  is a small constant and  $T_k$  is the temperature in iteration  $k$ . Regarding the acceptance function, we use the Metropolis acceptance function, where the probability to accept a new solution  $P_{\text{acceptance}}$  depends on the difference between the current accepted solution's objective value  $f(\text{old})$  and the new neighboring solution's objective value  $f(\text{new})$  ( $\Delta = f(\text{new}) - f(\text{old})$ ) (Metropolis et al. 1953). The Metropolis function is defined as:

$$P_{\text{acceptance}} = \begin{cases} \exp\left(-\frac{\Delta}{T_k}\right) & \text{if } \Delta > 0 \\ 1 & \text{if } \Delta \leq 0 \end{cases}$$

In the first iteration, the algorithm is initiated via a start temperature. The algorithm terminates when the temperature of an iteration falls below the predetermined end temperature. The required parameters are defined in Sect. 6.4 on the computational experiments with the simulated annealing algorithm.

The procedure can be summarized as:

1. Find an initial solution using the construction heuristic.
2. Find a neighboring solution by randomly choosing and applying an operator to alter the solution.
3. If the neighboring solution is invalid, set this move to tabu and repeat Step 2. Otherwise, reset the tabu list and continue with Step 4.
4. Accept or reject the solution based on the acceptance function.
5. Proceed with the next iteration, taking the cooling function into account.
6. Terminate the heuristic if the temperature falls below end temperature. Report the best solution. Otherwise, start again at Step 2.

## 6 Computational experiments and results

In this section, computational experiments are conducted to evaluate the effectiveness of the proposed approach. Section 6.1 introduces the design of the study regarding the storage structure, the crane, the coils, and operational settings. Section 6.2 justifies the alternative planning approaches used for comparison and explains the computational setup. The results of the exact approach are presented in Sect. 6.3, while the results of the simulated annealing algorithm are presented in Sect. 6.4.

### 6.1 Instance generation and data

Reasonable assumptions about the storage and the crane must be made. Regarding the storage, the indexing of places and the distances within the storage must be specified. Therefore, each place  $k$  is characterized by its row ( $r_k$ ), position ( $p_k$ ), and layer ( $l_k$ ).  $d^1$  defines the distance between rows,  $d^2$  the distance between positions, and  $d^3$  the distance between layers. A detailed overview of the storage structure is depicted in Fig. 6.

Regarding the crane, the crane travel times between places ( $t_{kq}^{\text{empty}}$  and  $t_{kq}^{\text{load}}$ ) and the associated energy consumption ( $E_{kq,a}^{\text{load}}$  and  $E_{kq}^{\text{empty}}$ ) must be specified. First, the travel time between places  $k$  and  $q$  is justified. Portal and trolley of gantry cranes can move simultaneously, i.e., the travel time depends on the maximum of the two movement axes (Chebyshev metric). These movements require the hook at lifting height which is identified by the corresponding layer  $l_H$ . However, for safety reasons neither portal nor trolley movement may co-occur during hook movements. The velocity of the movements depends on their direction. Different velocities are defined for the portal movement ( $v^1$ ), trolley movement ( $v^2$ ), and hook movement ( $v^3$ ). Therefore, the travel time for an empty movement between

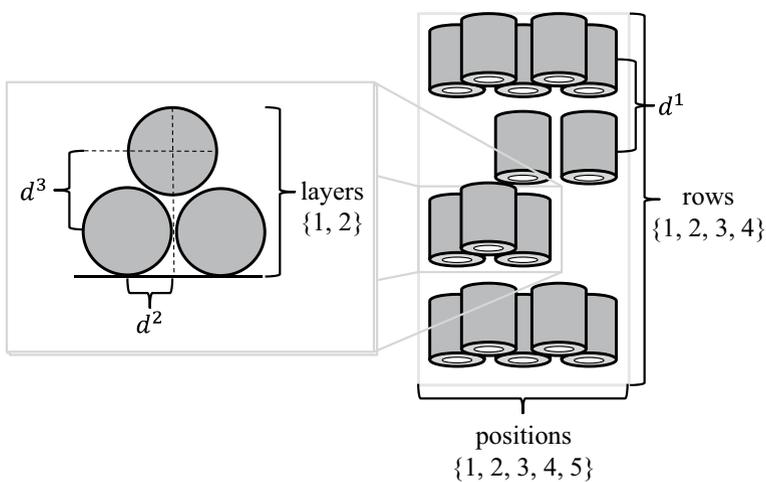


Fig. 6 Structure of the steel coil storage, specified for the set of small test instances

two places  $(k, q)$  results to  $t_{kq}^{\text{empty}} = \max\left(\frac{d^1 \cdot |r_k - r_q|}{v^1}, \frac{d^2 \cdot |p_k - p_q|}{v^2}\right) + \frac{d^3 \cdot (l_H - l_k)}{v^3} + \frac{d^3 \cdot (l_H - l_q)}{v^3}$ .

Compared to an empty move, a loaded move requires an additional closing and opening of the hook for pickup and dropping of the coil. The associated time is referred to as  $\mu$  ( $t_{kq}^{\text{load}} = t_{kq}^{\text{empty}} + \mu$ ). For the study, we assume the time for pickup and dropping of coils as  $\mu = 35$  s. We assume the distances to  $d^1 = 2.2$  m,  $d^2 = 0.8$  m, and  $d^3 = 1.0$  m, and the lifting height to 5 m. Further, velocities are  $v^1 = 1.667$  m/s,  $v^2 = 0.833$  m/s, and  $v^3 = 0.2$  m/s.

Second, energy consumption requires justification. Different authors assessed (gantry) crane energy consumption in different settings, which can serve to parameterize the energy consumption of crane movements (Alasali et al. 2019; Labitzke 2011; Papaioannou et al. 2017; Wu and Xia 2014). We base our parameters on the findings of Labitzke (2011). Accordingly, the energy consumption of an empty move  $E_{kq}^{\text{empty}}$  considers lifting and lowering the empty hook and portal and trolley movements between the places  $k$  and  $q$ . The energy consumption  $E_{kq,a}^{\text{load}}$  considers lifting and lowering coil  $a$  and portal and trolley movements between the places  $k$  and  $q$  depending on the weight  $m_a$  of each coil  $a$ .

The operational settings we consider are derived from applications observed in practice. A planning horizon of 120 min is considered. The time windows of the incoming coils do not overlap. We refer to an application where a production stage precedes the coil storage, in which a refinement process is carried out on the coils. These facilities typically require the refined coil to be retrieved by the crane from the facility (i.e., it needs to be stored) before the next coil completes its refinement. The time windows of the outgoing coils overlap with other outgoing and incoming coils, requiring order sequencing.

To provide insights into the structure of our problem and to derive implications for practice, nine different scenarios are considered varying the *initial occupancy levels* and the *time window length for outgoing coils*. These scenarios aim to examine the effects of higher or lower flexibility in crane scheduling, allowing decision-makers to assess the extent to which efforts to increase the available flexibility may be beneficial. To this end, we vary the initial occupancy level of the storage between 30, 50, and 70%. Further, we vary the length of the time windows for outgoing coils to 10, 20, and 30 min, respectively. The setup of the scenarios is summarized in Table 2.

We construct two sets of instances of different size. The small instances consist of a storage area with 4 rows with 5 places each in 2 layers, resulting in 20 storage places. The large instances consist of a storage area with 20 rows with 25 places each in 2 layers, resulting in 500 storage places. The large instances are of a size found in real-world industrial applications. The small instances serve for testing the solution procedures. For each of the resulting scenarios, 20 small and large instances are generated randomly.

While benchmark data have been published for other crane scheduling problems (Meisel and Bierwirth 2011; Vallada et al. 2023), those datasets differ in the structure and dimension of the storage and lack product weight data and the energy consumption of crane movements. To facilitate future research efforts specifically for

**Table 2** Scenario parameters and identifiers

Initial occupancy level			
	30%	50% (initially placed coils from the 30% scenario + additional coils)	70% (initially placed coils from the 50% scenario + additional coils)
<i>Time window length for outgoing coils</i>			
<b>10 min</b>	CS <sub>30%10 min</sub>	CS <sub>50%10 min</sub>	CS <sub>70%10 min</sub>
<b>20 min</b>	CS <sub>30%20 min</sub>	CS <sub>50%20 min</sub>	CS <sub>70%20 min</sub>
<b>30 min</b>	CS <sub>30%30 min</sub>	CS <sub>50%30 min</sub>	CS <sub>70%30 min</sub>
For all scenarios:			
● The identical eight incoming coils with fixed time windows and the same eight outgoing coils with a fixed deadline			
● 20 storage places (small instances)/500 storage places (large instances)			
For scenarios with different time window lengths for outgoing coils:			
● 20 instances for each initial occupancy level, where the initially placed coils are assigned in a random order to a random storage place			
● The place assignment stays identical between scenarios with different time window length, i.e., only the time window length for outgoing coils changes			

crane scheduling in steel coil storages, we provide the data for all instances as supplementary material to this article, which is available Open Access under Weckenborg et al. (2023).

## 6.2 Alternative planning approaches and setup

We compare the energy-oriented approach with established approaches from the literature to gain insight into the effects of energy-oriented crane scheduling and to analyze its benefits. In scientific literature, mainly time-oriented objectives are addressed (e.g., minimizing cycle time, makespan, or tardiness). As time windows apply for the orders in steel coil warehouses, these time-oriented objectives cannot reasonably be used in this setting. Therefore, we consult alternative planning approaches for an appropriate comparison with the energy-oriented approach.

In place assignment in container yards, minimizing the *number of reshuffles* is the most common objective function (Lehnfeld and Knust 2014). As reshuffles require additional crane movements and thus consume additional energy, minimizing the number of reshuffles reduces energy consumption. Therefore, we use this as the first alternative planning approach for comparison.

In simultaneous place assignment and sequencing of crane movements in container yards, minimizing *the crane travel times* was proposed Galle et al. (2018). As cranes consume energy during their travel, minimizing crane travel times reduces energy consumption. Besides estimating the energy consumption, the crane travel times may also serve to anticipate the crane utilization as another operative performance indicator. Therefore, this is the second alternative planning approach. To consider the two alternative planning approaches, the objective function of the

mathematical model is adapted. Therefore, the energy-oriented objective function (1) is exchanged with the alternative objective function (27) to minimize the total number of reshuffles ( $\text{res}_{\min}$ ) and (28) to minimize the total crane travel time ( $\text{ctt}_{\min}$ ), respectively. The remainder of the model formulation remains unchanged. This ensures that the results for the three planning approaches are generated *ceteris paribus*, facilitating their valid comparison.

$$\text{res}_{\min} : \min \sum_{k \in \Psi} \sum_{q \in \Psi} \sum_{s \in S} \sum_{a \in A} W_{kq,a}^s \quad (27)$$

$$\text{ctt}_{\min} : \min \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{s \in S} \sum_{a \in A} (W_{kq,a}^s \cdot t_{kq}^{\text{load}}) + \sum_{k \in \Phi} \sum_{q \in \Phi} \sum_{s \in S} (V_{kq}^s \cdot t_{kq}^{\text{empty}}) \quad (28)$$

### 6.3 Analysis of MILP results

This section illustrates the results of energy-oriented crane scheduling ( $\text{ener}_{\min}$ ) compared to the alternative planning approaches  $\text{res}_{\min}$  and  $\text{ctt}_{\min}$ . To this end, we detail each approach's energy consumption and crane travel times. Finally, the computational time required for the approaches is provided. The analysis is based on the 180 small instances solved once for each planning approach using the MILP. The resulting energy consumption of the alternative planning approaches is calculated ex-post based on the determined crane movements. The models are implemented in Python 3.10.4 and solved using Gurobi 9.5.0 on machines with 64 GB RAM and eight threads of an Intel Xeon Platinum 8180 CPU at 2.5 GHz. All instances were solved to optimality.

First, we look at the energy consumption. Using the energy-oriented approach  $\text{ener}_{\min}$ , the energy consumption can be reduced by 15.92% on average (22.34% best case, 5.97% worst case) compared to  $\text{res}_{\min}$ , and 6.66% on average (11.24% best case, 1.96% worst case) compared to  $\text{ctt}_{\min}$ . Therefore, a significant improvement in energy consumption can be achieved using the energy-oriented approach. The consumed energy is affected by the initial storage occupancy level and the time window lengths of outgoing coils. For  $\text{ener}_{\min}$ , an increase in time windows length from 10 to 20 min lowers the average energy consumption by 0.8%. A further increase from 20 to 30 min lowers the average energy consumption by an additional 1.8%. A reduction of the initial occupancy level from 70 to 50% results in energy savings of 3.8–4.0%. A further reduction to 30% reduces the energy consumption by another 0.0–0.7%. Regarding minimizing the crane travel times,  $\text{ctt}_{\min}$  profits in the same way from longer time windows as  $\text{ener}_{\min}$ . This is due to longer time windows enabling the planning model to shift the sequence so that retrieval operations free storage places such that the subsequent storing operations find beneficial places regarding crane travel time and energy consumption. Assigning this freed place to the next storing operation may lower the energy consumption and crane travel time but does not impact the number of reshuffles. Consequently, on the one hand, the energy consumption of  $\text{res}_{\min}$  cannot consistently be reduced with longer time windows of outgoing coils. On the other hand, a lower initial occupancy level leads to more free

storage places. This lowers the average number of reshuffles from 1.17 for 70% initial occupancy levels to 0.35 for 50% and 0.05 for 30%. However, this effect is the same for the three approaches, i.e.,  $ener_{\min}$ ,  $ctt_{\min}$ , and  $res_{\min}$  achieve an identical and minimal number of reshuffles for all instances. Therefore, the results on minimizing the overall reshuffles are not reported in detail. Generally speaking, minimizing the number of reshuffles seems to be a proper preliminary objective function, as yielding a minimal number of reshuffles seems necessary to achieve solutions with minimal energy consumption or crane travel times. However, additional potential can be lifted using  $ener_{\min}$  and  $ctt_{\min}$ .

Please refer to Table 3 for detailed results in which we report the average and standard deviation of the energy consumption of the 20 instances in each scenario for all three objective functions. Further, we report on the average delta between the energy-oriented approach and the two alternative planning approaches, and the worst and best solution compared to the alternative planning approaches. In summary, energy-oriented crane scheduling can reduce the average energy consumption compared to the alternative planning approaches established in literature by 5.36–17.44%.

Second, we relate to crane travel times the different approaches yield (cf. Table 4). Naturally,  $ctt_{\min}$  achieves the best results. However, the crane travel time of  $ener_{\min}$  increases only by 2.12% on average compared to  $ctt_{\min}$ . Therefore, crane utilization is only slightly negatively affected while a relevant extent of energy can be saved using  $ener_{\min}$ . Additionally, compared to  $res_{\min}$ , the average crane travel time of  $ener_{\min}$  is 6.43% lower, showing that the energy-oriented approach is superior to minimizing reshuffles in energy consumption and crane travel time. Please note that it is ensured that the orders are fulfilled without tardiness in all considered approaches, i.e., a higher crane utilization does not affect order fulfillment.

Finally, we relate to the computational time required to find the optimal solutions and prove their optimality (cf. Table 5). The lowest time is required by  $res_{\min}$ .  $ener_{\min}$  takes on average 412% the time of  $res_{\min}$  and 69% of the time of  $ctt_{\min}$ . Comparing the solution times of  $ctt_{\min}$  and  $ener_{\min}$  across the scenarios with different time window lengths for outgoing coils reveals major deviations in computational time. While in scenarios with 10 min time window length for outgoing coils, the computation of  $ener_{\min}$  takes 129% of the time of  $ctt_{\min}$ , it takes 52% of the time of  $ctt_{\min}$  for scenarios with 30 min time windows. This is mainly caused by a few instances taking up to 14 days to solve optimally for  $ctt_{\min}$ , indicated by the high standard deviation. The average time to find a feasible solution using  $ener_{\min}$  is 2.4 h and its average gap is 9.98%. Only in 37.8% of the instances, a feasible solution is found in less than an hour.

In conclusion, the energy-oriented crane scheduling approach  $ener_{\min}$  reduces the energy consumption of crane movements by 15.92 and 6.66% on average compared to  $res_{\min}$  and  $ctt_{\min}$ , respectively. The crane travel times resulting from  $ener_{\min}$  are only slightly higher than those of  $ctt_{\min}$ . However, all orders are provided without tardiness. All three approaches yield an equal number of reshuffles for each instance. The required computational times of  $ener_{\min}$  are significantly higher than those of  $res_{\min}$ .

**Table 3** Comparison of the energy consumption

Scenario	Minimize energy consumption					Avg. (std.) of $\text{res}_{\min}$ [kWh]	Avg. (std.) of $\text{ctf}_{\min}$ [kWh]
	Avg. $\Delta$ to $\text{res}_{\min}$ [%]	Worst/best $\Delta$ to $\text{res}_{\min}$ [%]	Avg. $\Delta$ to $\text{ctf}_{\min}$ [%]	Worst/best $\Delta$ to $\text{ctf}_{\min}$ [%]	Avg. (std.) of $\text{res}_{\min}$ [kWh]		
CS <sub>30%</sub> 10 min	-14.14	-6.13 / -18.39	-6.60	-4.73 / -7.56	40.53 (1.82)	37.26 (0.50)	
CS <sub>30%</sub> 20 min	-14.96	-7.27 / -19.52	-6.54	-5.03 / -7.78	40.51 (2.05)	36.86 (0.43)	
CS <sub>30%</sub> 30 min	-16.66	-8.88 / -20.93	-5.36	-2.54 / -8.00	40.46 (1.98)	35.63 (0.71)	
CS <sub>50%</sub> 10 min	-16.06	-5.97 / -19.33	-6.97	-4.88 / -9.64	41.47 (1.57)	37.42 (1.27)	
CS <sub>50%</sub> 20 min	-16.20	-7.44 / -20.59	-7.19	-4.56 / -9.96	41.29 (2.28)	37.28 (1.32)	
CS <sub>50%</sub> 30 min	-17.44	-10.97 / -21.22	-7.01	-3.82 / -11.24	41.16 (1.68)	36.54 (1.31)	
CS <sub>70%</sub> 10 min	-14.84	-6.87 / -19.12	-6.59	-1.96 / -10.20	42.58 (2.95)	38.82 (1.91)	
CS <sub>70%</sub> 20 min	-15.80	-9.03 / -20.51	-6.84	-4.13 / -10.01	42.71 (2.19)	38.60 (1.96)	
CS <sub>70%</sub> 30 min	-17.15	-10.52 / -22.34	-6.79	-2.60 / -10.43	42.73 (2.55)	37.98 (1.92)	

Average (avg.) and standard deviation (std.) of the consumed energy for the included instances in kWh, deviation of the consumed energy of  $\text{ener}_{\min}$  compared to the alternative planning approaches on average of the included instances in % (avg.  $\Delta$ ), lowest and highest deviation of the consumed energy of  $\text{ener}_{\min}$  compared to the alternative planning approaches among the included instances in % (worst/best  $\Delta$ )

**Table 4** Comparison of the crane travel time

Scenario	Minimize energy consumption				Avg. (std.) of $res_{min}$ [h]	Avg. (std.) of $ctt_{min}$ [h]	Avg. (std.) of $ct_{min}$ [h]
	Avg. (std.) of $ener_{min}$ [h]	Avg. $\Delta$ to $res_{min}$ [%]	Worst/best $\Delta$ to $res_{min}$ [%]	Avg. $\Delta$ to $ctt_{min}$ [%]			
CS <sub>30%</sub> 10 min	0.508 (0.009)	-6.10	0.19 / -11.74	2.42	4.87/0.81	0.541 (0.017)	0.496 (0.007)
CS <sub>30%</sub> 20 min	0.511 (0.008)	-5.02	0.59 / -8.35	3.23	4.08/1.43	0.538 (0.019)	0.495 (0.007)
CS <sub>30%</sub> 30 min	0.501 (0.007)	-7.22	-1.94 / -10.11	2.04	3.70/1.23	0.540 (0.018)	0.491 (0.007)
CS <sub>50%</sub> 10 min	0.510 (0.018)	-6.76	0.40 / -9.88	2.41	3.89/1.22	0.547 (0.019)	0.498 (0.017)
CS <sub>50%</sub> 20 min	0.510 (0.018)	-6.76	0.00 / -9.31	2.41	4.28/0.82	0.547 (0.025)	0.498 (0.017)
CS <sub>50%</sub> 30 min	0.503 (0.017)	-7.37	-2.05 / -11.11	1.21	2.26/0.41	0.543 (0.018)	0.497 (0.017)
CS <sub>70%</sub> 10 min	0.532 (0.032)	-5.51	2.37 / -8.94	2.31	6.57/0.41	0.563 (0.038)	0.520 (0.030)
CS <sub>70%</sub> 20 min	0.528 (0.030)	-6.05	-1.42 / -9.33	1.93	5.34/0.21	0.562 (0.029)	0.518 (0.028)
CS <sub>70%</sub> 30 min	0.523 (0.029)	-7.10	-2.13 / -9.78	1.16	3.09/0.21	0.563 (0.032)	0.517 (0.028)

Average (avg.) and standard deviation (std.) of the crane travel time for the included instances in h, deviation of the crane travel time of  $ener_{min}$  compared to the alternative planning approaches on average of the included instances in % (avg.  $\Delta$ ), highest and lowest deviation of the crane travel time of  $ener_{min}$  compared to the alternative planning approaches among the included instances in % (worst/best  $\Delta$ )

**Table 5** Time required to find an optimal solution in different scenarios based on the objective function

Scenario	Minimize energy consumption			Avg. (std.) of res <sub>min</sub> [h]			Avg. (std.) of res <sub>min</sub> [h]		
	Avg. (std.) of ener <sub>min</sub> [h]	Avg. Δ to res <sub>min</sub> [%]	Worst/best Δ to res <sub>min</sub> [%]	Avg. Δ to res <sub>min</sub>	Worst/best Δ to res <sub>min</sub> [%]	Avg. Δ to ctt <sub>min</sub> [%]	Worst/best Δ to ctt <sub>min</sub> [%]	Avg. (std.) of res <sub>min</sub> [h]	Avg. (std.) of ctt <sub>min</sub> [h]
CS <sub>30%</sub> 10 min	2.86 (1.14)	822.58	1591.34/270.00	18.18	1215.68 / -56.58	0.31 (0.04)	2.42 (2.97)		
CS <sub>30%</sub> 20 min	4.10 (1.61)	446.67	1934.94 / -8.24	-49.57	1071.45 / -89.25	0.75 (1.66)	8.13 (6.60)		
CS <sub>30%</sub> 30 min	2.48 (1.04)	463.64	792.37/202.79	-46.90	659.91 / -85.50	0.44 (0.04)	4.67 (7.33)		
CS <sub>50%</sub> 10 min	3.19 (2.49)	258.43	1195.45/34.39	0.63	1307.05 / -89.87	0.89 (1.66)	3.17 (9.70)		
CS <sub>50%</sub> 20 min	5.87 (5.02)	789.39	4644.49/131.49	341.35	1726.12 / -22.01	0.66 (0.25)	1.33 (2.11)		
CS <sub>50%</sub> 30 min	3.79 (4.06)	112.92	1206.64 / -67.87	-32.08	1422.53 / -87.76	1.78 (3.25)	5.58 (14.33)		
CS <sub>70%</sub> 10 min	10.18 (8.52)	297.66	5167.31 / -73.76	46.26	1669.65 / -70.19	2.56 (4.88)	6.96 (11.31)		
CS <sub>70%</sub> 20 min	13.15 (9.22)	500.46	4623.24/36.20	-49.13	2059.29 / -88.12	2.19 (4.96)	25.85 (73.80)		
CS <sub>70%</sub> 30 min	13.23 (13.49)	181.49	5164.41 / -81.93	-51.98	2204.31 / -88.75	4.70 (9.62)	27.55 (50.92)		

Average (avg.) and standard deviation (std.) of the computational time for the included instances in h, deviation of the computational time of ener<sub>min</sub> compared to the alternative planning approaches on average of the included instances in % (avg. Δ), highest and lowest deviation of the computational time of ener<sub>min</sub> compared to the alternative planning approaches among the included instances in % (worst/best Δ)

## 6.4 Analysis of the simulated annealing algorithm

As shown in the previous section, an energy-oriented approach to the crane scheduling problem in steel coil storage can be considered advantageous compared to other planning approaches. However, the required computational times of the MILP are not suitable for practical applications. In this section we evaluate the performance of the simulated annealing algorithm. First, the parameters for the SA are configured. Second, using the small instances previously solved to optimality, the SA results are compared to MILP results regarding computational times and solution quality. Finally, results on the large instances generated with the SA are reported. The SA is implemented in Python and the experiments are executed on the same system as the MILP experiments.

The required parameters for the SA are selected based on preliminary studies. The start temperature is chosen to be  $t_0 = 25$ . The end temperature is set to  $t_{\text{end}} = 0.01$  indicating a compromise between the quality of solutions and computational time. For the cooling function, we set  $\beta = 0.01$ , resulting in 10,000 iterations of the SA for the given set of parameters. The last parameters account for choosing the operator to alter the solution. In each iteration, one operator is randomly chosen. The operators *sequence shift* and *place shift* occur in 75 and 25% of iterations, respectively. We use a multi-start approach for the SA, in which the first run starts with the solution of the construction heuristic, while the second and third runs start with a random solution of the previous run. The solution times reported in the following account for the sum of the three runs, while the information on the gap refers to the best solution found among the three runs.

With this set of parameters, the SA solves the small instances in 59.7 s on average, with an average gap to the optimal solution of 2.59%. Solving the large instances took an average of 302.8 s. Detailed data on the solution times and quality is provided in Table 6.

This indicates that the SA finds close-to-optimal solutions for the small instances. Large instances of a size encountered in practice can be solved in a reasonable time. Please note that the instances account for a planning horizon of two hours. Based on the feedback from our collaborator in practice, the reported solution times are suitable for practical application.

## 7 Managerial implications

In this section, insights into the relevance of the findings for practice are derived. Those are threefold: an estimation of the impact of energy-oriented crane scheduling on the bottom line of a steel company, insights into the contribution of flexibility to efficient crane scheduling, and a structural analysis of the solutions to highlight patterns enabling energy savings.

First, an estimation of the impact of the findings on the bottom line of a steel company is desired. To this end, we discussed the approach of energy-oriented crane scheduling with a German steel manufacturer and transferred the results to a real-world application. Our collaborators move 1750 coils per day on average using 26

**Table 6** Solution quality and times for the SA approach

Scenario	Small instances			Large instances
	Avg. (std.) Gap [%]	min/max Gap [%]	Avg. (std.) CPU [s]	Avg. (std.) CPU [s]
CS <sub>30%10 min</sub>	1.96 (0.30)	1.41/2.61	59.7 (2.7)	189.5 (24.0)
CS <sub>30%20 min</sub>	2.62 (0.44)	1.66/3.30	55.5 (2.6)	158.4 (17.2)
CS <sub>30%30 min</sub>	4.34 (0.77)	2.36/5.37	53.5 (2.4)	153.0 (22.7)
CS <sub>50%10 min</sub>	2.33 (0.81)	1.26/4.25	62.0 (4.5)	323.2 (32.7)
CS <sub>50%20 min</sub>	2.35 (0.75)	1.30/3.62	56.9 (3.1)	233.6 (34.4)
CS <sub>50%30 min</sub>	3.06 (0.80)	2.03/5.20	54.1 (3.6)	222.1 (27.4)
CS <sub>70%10 min</sub>	2.22 (1.27)	0.41/5.74	72.3 (10.3)	526.1 (90.5)
CS <sub>70%20 min</sub>	2.14 (1.18)	0.32/4.77	63.6 (7.7)	475.1 (72.4)
CS <sub>70%30 min</sub>	2.30 (0.67)	0.93/3.51	59.6 (6.5)	444.1 (84.1)

Average (avg.) and standard deviation (std.) of the gap to optimal solutions of small instances in %, minimal and maximal gap of the instances contained in this scenario in % (min/max Gap), average solution time of the instances contained in this scenario in seconds

gantry cranes. For the instances used in our computational experiments in Sect. 6.3, we found the energy-oriented approach to consume less energy than the alternative planning approaches by 2–22%. If this range also applies to instances of real-world size, this could result in savings between 8000–103,000 € annually, based on the average German electricity costs of 26.5 ct/kWh for industrial customers in 2023 (BDEW 2024). Please note that these savings are based on comparing the energy-oriented approach to the optimal results of the two alternative planning approaches. In the practice of our collaborator, crane scheduling is mainly conducted by human crane operators, and algorithms are barely found. Therefore, we assume energy-oriented crane scheduling might have an even more significant impact in practice. Currently, we are working on the roll-out of our approach to practice.

Second, the computational experiments allow anticipating the impact of flexibility in the operational setting of manufacturers. As described in the previous results, increasing the length of the time windows can reduce the average energy consumption by 0.8% for an increase from 10 to 20 min and an additional 1.8% for an increase from 20 to 30 min (cf. Table 3). Therefore, on the one hand, longer time windows may serve as a straightforward measure to yield minor energy savings if the operational setting of the manufacturers allows adapting them. On the other hand, a reduction of inventory levels from 70 to 50% initial occupancy level results in energy savings of 3.8–4.0%. However, a further reduction to 30% reduces the energy consumption only by another 0.0–0.7%, i.e., decreasing marginal returns apply (cf. Table 3). Consequently, maintaining flexibility on the operational level may facilitate a further reduction of energy consumption. However, manufacturers must carefully evaluate the potential economic tradeoffs between maintaining flexibility and the potential improvements.

Third, we facilitate the readers' understanding of the patterns enabling energy savings for crane scheduling in steel coil storages. As previously reported in the results, all three approaches successfully avoid reshuffles. As reshuffles represent

non-value-adding operations, it is easily understood that they should be avoided. Consequently, all three approaches are characterized by the same number of loaded and empty movements. However, further insights can be concluded from the structural analysis of the solutions. Accordingly, in energy-oriented solutions, *travel in multiple directions* to fulfill an order is avoided. A causal relationship between overall energy consumption and the number of movements differentiated by directions seems plausible, given the high mass of the portal and the trolley. Consequently, overcoming the cranes' inertia significantly contributes to energy consumption. Therefore, crane operators may want to avoid initializing movements in multiple directions for an order.

## 8 Conclusions and outlook

In this article, an energy-oriented approach for crane scheduling is addressed for the first time and applied to steel coil storages. The developed mixed-integer linear programming model for energy-oriented crane scheduling is compared to popular alternatives from the literature that aim to minimize the number of reshuffles and the crane travel time. The analyses indicate that the energy-oriented approach decreases the crane operations' energy consumption by 5.4–17.4% at comparable crane utilization and without tardiness compared to the alternative planning approaches. The consideration of the specific energy consumption of the crane movements within planning approaches for crane scheduling can therefore be recommended. However, from a methodological standpoint, the mixed-integer linear programming model does not generate results in an adequate computational time. Therefore, we develop a simulated annealing algorithm which finds good solutions for instances of real-world size within a reasonable amount of time.

This article is dedicated to crane scheduling in steel coil storages, where a triangular stacking pattern must be considered. However, the idea of energy-oriented crane scheduling can be transferred toward general settings based on our findings. This is particularly relevant, for example, for scheduling yard cranes, and seems like a promising way to reduce the energy consumption of operations in container terminals.

**Acknowledgements** The authors wish to thank the editor and the three anonymous reviewers for their valuable suggestions, which significantly helped to improve a former version of this manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. No funding was received to assist with the preparation of this manuscript. The authors have no relevant financial or non-financial interests to disclose.

## Declarations

**Ethical approval** (1) This material is the authors' own original work, which has not been previously published elsewhere. (2) The paper is not currently being considered for publication elsewhere. (3) The paper reflects the authors' own research and analysis in a truthful and complete manner. (4) The results are appropriately placed in the context of prior and existing research. (5) All sources used are properly disclosed

(correct citation). Literally copying of text must be indicated as such by using quotation marks and giving proper reference. (6) All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Alasali F, Haben S, Holderbaum W (2019) Energy management systems for a network of electrified cranes with energy storage. *Int J Electr Power Energy Syst* 106:210–222. <https://doi.org/10.1016/j.ijepes.2018.10.001>
- BDEW (2024) Annual electricity prices (including electricity tax) for industrial businesses in Germany from 1998 to 2023 (in euro cents per kilowatt hour). <https://www.statista.com/statistics/1050448/industrial-electricity-prices-including-tax-germany/>. Accessed 25 Jan 2024
- Bierwirth C, Meisel F (2010) A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur J Oper Res* 202:615–627. <https://doi.org/10.1016/j.ejor.2009.05.031>
- Bierwirth C, Meisel F (2015) A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur J Oper Res* 244:675–689. <https://doi.org/10.1016/j.ejor.2014.12.030>
- Boysen N, Stephan K (2016) A survey on single crane scheduling in automated storage/retrieval systems. *Eur J Oper Res* 254:691–704. <https://doi.org/10.1016/j.ejor.2016.04.008>
- Boysen N, Briskorn D, Meisel F (2017) A generalized classification scheme for crane scheduling with interference. *Eur J Oper Res* 258:343–357. <https://doi.org/10.1016/j.ejor.2016.08.041>
- Dohn A, Clausen J (2010) Optimising the slab yard planning and crane scheduling problem using a two-stage heuristic. *Int J Prod Res* 48:4585–4608. <https://doi.org/10.1080/00207540902998331>
- Galle V, Barnhart C, Jaillet P (2018) Yard crane scheduling for container storage, retrieval, and relocation. *Eur J Oper Res* 271:288–316. <https://doi.org/10.1016/j.ejor.2018.05.007>
- Gharehgozli AH, Roy D, de Koster R (2016) Sea container terminals: new technologies and OR models. *Marit Econ Logist* 18:103–140. <https://doi.org/10.1057/mel.2015.3>
- Heshmati S, Toffolo TA, Vancroonenburg W, Vanden Berghe G (2019) Crane-operated warehouses: integrating location assignment and crane scheduling. *Comput Ind Eng* 129:274–295. <https://doi.org/10.1016/j.cie.2019.01.039>
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680. <https://doi.org/10.1126/science.220.4598.671>
- Labitzke N (2011) Wertorientierte Simulation zur taktischen Planung logistischer Prozesse der Stahlherstellung. Gabler, Wiesbaden
- Lee D-H, Cao Z, Meng Q (2007) Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *Int J Prod Econ* 107:115–124. <https://doi.org/10.1016/j.ijpe.2006.08.003>
- Lehnfeld J, Knust S (2014) Loading, unloading and premarshalling of stacks in storage areas: survey and classification. *Eur J Oper Res* 239:297–312. <https://doi.org/10.1016/j.ejor.2014.03.011>
- Liu S, Tang J, Song J (2006) Order-planning model and algorithm for manufacturing steel sheets. *Int J Prod Econ* 100:30–43. <https://doi.org/10.1016/j.ijpe.2004.10.002>
- Lu Q, Dessouky MM (2006) A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *Eur J Oper Res* 175:672–687. <https://doi.org/10.1016/j.ejor.2005.05.012>
- Lundy M, Mees A (1986) Convergence of an annealing algorithm. *Math Program* 34:111–124. <https://doi.org/10.1007/BF01582166>

- Maschietto GN, Ouazene Y, Ravetti MG, de Souza MC, Yalaoui F (2017) Crane scheduling problem with non-interference constraints in a steel coil distribution centre. *Int J Prod Res* 55:1607–1622. <https://doi.org/10.1080/00207543.2016.1193249>
- Meisel F, Bierwirth C (2011) A unified approach for the evaluation of quay crane scheduling models and algorithms. *Comput Oper Res* 38:683–693. <https://doi.org/10.1016/j.cor.2010.08.001>
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *J Chem Phys* 21:1087–1092. <https://doi.org/10.1063/1.1699114>
- Papaioannou V, Pietrosanti S, Holderbaum W, Becerra VM, Mayer R (2017) Analysis of energy usage for RTG cranes. *Energy* 125:337–344. <https://doi.org/10.1016/j.energy.2017.02.122>
- Peng G, Wu Y, Zhang C, Shen W (2021) Integrated optimization of storage location assignment and crane scheduling in an unmanned slab yard. *Comput Ind Eng* 161:107623. <https://doi.org/10.1016/j.cie.2021.107623>
- Puttkammer K, Wichmann MG, Spengler TS (2016) A GRASP heuristic for the hot strip mill scheduling problem under consideration of energy consumption. *J Bus Econ* 86:537–573. <https://doi.org/10.1007/s11573-015-0783-3>
- Rei RJ, Kubo M, Pedroso JP (2008) Simulation-based optimization for steel stacking. Modelling, computation and optimization in information systems and management sciences: second international conference MCO 2008, Metz, France - Luxembourg, September 8–10, 2008. Springer, Berlin, pp 254–263
- Stahlbock R, Voß S (2007) Operations research at container terminals: a literature update. *Or Spectrum* 30:1–52. <https://doi.org/10.1007/s00291-007-0100-9>
- Steenken D, Voß S, Stahlbock R (2004) Container terminal operation and operations research—a classification and literature review. *Or Spectrum* 26:3–49. <https://doi.org/10.1007/s00291-003-0157-z>
- Tang L, Liu J, Rong A, Yang Z (2001) A review of planning and scheduling systems and methods for integrated steel production. *Eur J Oper Res* 133:1–20. [https://doi.org/10.1016/S0377-2217\(00\)00240-X](https://doi.org/10.1016/S0377-2217(00)00240-X)
- Tang L, Xie X, Liu J (2014) Crane scheduling in a warehouse storing steel coils. *IIE Trans* 46:267–282. <https://doi.org/10.1080/0740817X.2013.802841>
- Vallada E, Belenguer JM, Villa F, Alvarez-Valdes R (2023) Models and algorithms for a yard crane scheduling problem in container ports. *Eur J Oper Res* 309:910–924. <https://doi.org/10.1016/j.ejor.2023.01.047>
- Wang X, Zhou M, Zhao Q, Liu S, Guo X, Qi L (2021) A branch and price algorithm for crane assignment and scheduling in slab yard. *IEEE Trans Autom Sci Eng* 18:1122–1133. <https://doi.org/10.1109/TASE.2020.2996227>
- Weckenborg C, Oetjegerdes P, Spengler TS (2023) Supplementary material to “Energy-oriented crane scheduling in a steel coil storage”. *Flex Serv Manuf J*. Zenodo. <https://doi.org/10.5281/zenodo.8110281>
- Wichmann MG, Spengler TS (2015) Slab scheduling at parallel continuous casters. *Int J Prod Econ* 170:551–562. <https://doi.org/10.1016/j.ijpe.2015.08.003>
- Wu Z, Xia X (2014) Energy efficiency of overhead cranes. *IFAC Proc Vol* 47:19–24. <https://doi.org/10.3182/20140824-6-ZA-1003.00961>
- Xie X, Zheng Y, Li Y (2014) Multi-crane scheduling in steel coil warehouse. *Expert Syst Appl* 41:2874–2885. <https://doi.org/10.1016/j.eswa.2013.10.022>
- Xu Z, Zheng Z, Gao X (2021) Operation optimization of the steel manufacturing process: a brief review. *Int J Miner Metall Mater* 28:1274–1287. <https://doi.org/10.1007/s12613-021-2273-7>
- Yuan Y, Tang L (2017) Novel time-space network flow formulation and approximate dynamic programming approach for the crane scheduling in a coil warehouse. *Eur J Oper Res* 262:424–437. <https://doi.org/10.1016/j.ejor.2017.03.007>
- Zäpfel G, Wasner M (2006) Warehouse sequencing in the steel supply chain as a generalized job shop model. *Int J Prod Econ* 104:482–501. <https://doi.org/10.1016/j.ijpe.2004.10.005>
- Zhao G, Liu J, Tang L, Zhao R, Dong Y (2020) Model and heuristic solutions for the multiple double-load crane scheduling problem in slab yards. *IEEE Trans Autom Sci Eng* 17:1307–1319. <https://doi.org/10.1109/TASE.2019.2946196>

**Christian Weckenborg** is an assistant professor at the Institute of Automotive Management and Industrial Production at Technische Universität Braunschweig, Germany. He holds an MSc degree in industrial engineering and a PhD in business administration. His research focuses on contributions of technology-oriented management in production and logistics. To this end, operations research methods are applied to provide informed business decision support.

**Patrick Oetjegerdes** is a research associate at the Institute of Automotive Management and Industrial Production at Technische Universität Braunschweig, Germany. He holds an MSc degree in industrial engineering. His research focuses on the application of simulation and optimization methods, especially in the context of production and logistics planning in the steel industry.

**Thomas S. Spengler** is professor and director of the Institute of Automotive Management and Industrial Production at Technische Universität Braunschweig, Germany. He holds a PhD from Karlsruhe Institute of Technology, Germany. His research covers the development and implementation of techno-economic models and quantitative methods for decision support. His work has been published in a variety of academic journals. Since 2020, he has been a member of the German Science and Humanities Council.