LogForum

> Scientific Journal of Logistics <

2021, 17 (2), 243-251

http://doi.org/10.17270/J.LOG.2021.572

http://www.logforum.net p-ISSN 1895-2038 e-ISSN 1734-459X

ORIGINAL PAPER

MIXED-INTEGER PROGRAMMING FORMULATIONS FOR THE TRUCK SCHEDULING PROBLEM WITH FIXED OUTBOUND DEPARTURES

Pascal Wolff[1], Jiazhen Huo[1], Hans-Christian Pfohl[2]

1) Tongji University, Shanghai, **China** 2) Technical University of Darmstadt, Darmstadt, **Germany**

ABSTRACT. Background: Truck scheduling at cross-docking terminals has received much academic attention over the last three decades. A vast number of mixed-integer programming models have been proposed to assign trucks to dock-doors and time slots. Surprisingly, only a few models assume fixed outbound truck departures that are often applied in the less-than-truckload or small parcel and express delivery industry. To the best of our knowledge, none of these papers explore whether a discrete-time or continuous-time model formulation has a better computational performance. This paper attempts to close this research gap and tries to shed light on which type of formulation is advantageous. Therefore, a variant of the truck scheduling problem with fixed outbound departures is considered. This problem's objective is to find a feasible truck schedule that minimizes the number of delayed freight units.

Methods: We propose two model formulations for the described variant of the truck scheduling problem with fixed outbound departures. Specifically, the problem is formulated as a discrete-time and a continuous-time mixed-integer programming model.

Results: A computational experiment is conducted in order to assess the computational performance of the presented model formulations. We compare the discrete-time and continuous-time formulation in terms of both the solution quality and computational time.

Conclusions: The computational results show that the proposed discrete-time model formulation can solve problem instances of medium size to proven optimality within less than one minute. The continuous-time model formulation, on the other hand, can solve small instances to optimality. However, it requires longer solution times than the discrete-time formulation. Furthermore, it is unable to solve medium-sized instances within a 5-minute time limit. Thus, it can be summarized that the proposed discrete-time model formulation is clearly superior to the continuous-time model formulation.

Key words: cross-docking; truck scheduling; mixed-integer programming; logistics; optimization.

INTRODUCTION

Cross-docking is a warehousing concept where incoming shipments are unloaded, sorted, and (directly) transferred to outgoing trucks. It aims to synchronize inbound and outbound shipments to avoid lengthy storage times and reduce the order picking effort. By consolidating less-than-truckload (LTL) shipments, cross-docking can also yield transportation cost savings compared to traditional point-to-point deliveries. Cross-docking terminals can be found in many of today's retailing (e.g., Wal-Mart [Stalk et al. 1992]), parcel delivery (e.g., UPS [Forger 1995] or DHL [Boysen et al. 2013]), automotive (e.g., Toyota [Witt 1998] or Renault [Serrano et al. 2017]), and logistics service provider [Gue 1999] supply chains.

A vast number of strategic (e.g., location and layout of cross-docking terminals), tactical (e.g., transportation flow optimization), and operational (e.g., truck assignment or truck scheduling) cross-docking decision problems

Citation: Wolff P., Huo J., Pfohl H.-Ch., 2021. Mixed-integer programming formulations for the truck scheduling problem with fixed outbound departures. LogForum 17 (2), 243-251, http://doi.org/10.17270/J.LOG.2021.572

Received: 25.12.2020, Accepted: 10.03.2021, on-line: 12.04.2021.

have been studied in academic publications. Recent literature reviews are provided by Van Belle et al. [2012] and Buijs et al. [2014]. Especially truck scheduling, an operational decision problem in cross-docking terminals that deals with both assigning trucks to dock-doors and time slots, received much academic attention. Boysen and Fliedner [2010] and Ladier and Alpan [2016] provide an in-depth overview of the literature.

Many truck scheduling studies assume constraint-free outbound departures [e.g., Chmielewski et al. 2009, Serrano et al. 2017, Shakeri et al. 2012]. Under this assumption, outgoing trucks may only leave the cross-docking terminal after all freight units were loaded. However, this assumption could not be applicable in industries such as the LTL logistics industry or parcel delivery industry, which rely on fixed outbound departures in order to realize a smooth material flow in the transportation network [Ladier and Alpan 2016, Boysen et al. 2013]. Surprisingly, only a few studies consider a truck scheduling problem with fixed outbound departures. Minimizing the number of delayed product units is among the most frequently used performance indicators in truck scheduling models that consider fixed outbound departures. Existing truck scheduling models with fixed outbound departures can be classified into continuous-time [e.g., Boysen et al. 2013, Molavi et al. 2018] and discrete-time [e.g., Rahmandzadeh Tootkaleh et al. 2016, Tadumadze et al. 2019, Wolff et al. 2021] mixed-integer programs. Continuous-time models (CT) use a set of continuous decision variables to specify when trucks are processed. In these models, truck processing can start at any time within a truck's time window. Furthermore, the models often rely upon disjunctive (precedence) constraints in combination with precedence-based (binary) decision variables to express the processing sequence between pairs of trucks assigned to the same dock-door. Continuous-time model formulations are often characterized by a weak relaxation and large search trees as they often include many big-M formulations [Lamorgese and Mannino 2019]. Discrete-time model formulations (DT) were introduced to overcome this significant drawback. They discretize the planning horizon and use time-indexed (binary) decision variables that simultaneously indicate the truck-to-door assignment and the time a truck is processed. Discrete-time model formulations are usually characterized by stronger relaxations and lower bounds, and a larger number of decision variables than their continuous-time counterparts.

To the best of our knowledge, no paper that studies a variant of the truck scheduling problem with fixed outbound departures compared discrete-time and continuous-time model formulations regarding their computational performance. This paper attempts to close this research gap, as it may shed light on which type of formulation is advantageous. We propose both a discrete-time and a continuous-time mixed-integer programming formulation for a variant of the truck scheduling problem with fixed outbound departures. The model formulations are then compared in a computational experiment regarding their solution quality and computational time.

TRUCK SCHEDULING PROBLEM WITH FIXED OUTBOUND DEPARTURES

Model assumptions

This paper studies a variant of the truck scheduling problem with fixed outbound departures (TSFD). The general model assumptions can be summarized as follows:

- Inbound trucks and outbound trucks must be processed at inbound doors and outbound doors, respectively (exclusive service mode).
- The outbound truck departure times (and the truck-to-door assignments for outbound trucks) are given and known in advance (fixed outbound departures).
- Each inbound truck has a time window, defined through the truck release time and due date, in which truck processing must start.
- Only standardized freight units (e.g., pallets) are handled at the cross-docking terminal, and a sort-at-receiving protocol is applied [Bartholdi et al. 2008].

- An inbound truck's processing time includes the time for unloading all cargo from the inbound truck and transporting it to the associated outbound dock-doors. Thus, an inbound truck's processing time depends on the number of product units and the travel distance between inbound and outbound dock-doors [Van Belle et al. 2013, Wolff et al. 2021].
- A truck cannot leave the dock-door it is assigned to before it has been processed completely (no preemption).
- Cargo that arrives in the outbound area after loading operations of an associated outbound truck started is regarded as delayed cargo and postponed until the next departure to the same destination [Van Belle et al. 2012, Wolff et al. 2021].

In this setting, the objective is to find a feasible schedule for inbound trucks that leads to a minimum number of delayed products. Such a setting and goal is relevant in unit-load cross-docking platforms of logistics service providers or retailing companies. In the following, we present different mixed-integer programming (MIP) formulations for the TSFD.

Model formulations

The discrete-time formulation, denoted as TSFD-DT, uses the set of binary variables x_{idt} where $i \in I, d \in D, t \in T$. We set $x_{idt} = 1$ if inbound truck $i \in I$ is processed at dock-door $d \in D$ and processing starts in time interval $t \in T$. Moreover, a set of binaries y_{io} is used in order to signal if inbound truck i's cargo reaches the outbound area before loading of outbound truck o starts. When applying the notation summarized in Table 1, the TSFD-DT can be formulated as shown in Table 2.

Table 1. Notations for the discrete-time model formulation of the TSFD

Sets:	
I	Set of inbound trucks.
O	Set of outbound trucks.
D	Set of inbound doors.
T	Set of time intervals.
Parameters:	
r_i	Release time of inbound truck $i \in I$.
d_i	Due date of inbound truck $i \in I$.
d_o	Time when the processing of outbound truck $o \in O$ starts.
p_{id}	Processing time of inbound truck $i \in I$ at inbound dock-door $d \in D$.
f_{io}	Material flow between inbound truck $i \in I$ and outbound truck $o \in O$.
M	Big number.
Decision variables:	
x_{idt}	Binary decision variable: 1, if inbound truck $i \in I$ is assigned to door $d \in D$ and processing starts in time interval $t \in T$; 0, otherwise.
y_{io}	Binary decision variable: 1, if the processing of inbound truck $i \in I$ is finished after processing of outbound truck $o \in O$ starts; 0, otherwise.

Source: own table

Table 2. Discrete-time model for the TSFD

Objective function:		
$Minimize \sum_{i \in I} \sum_{o \in O} f_{io} \cdot y_{io}$		(1)
Constraints:		
$\sum_{d \in D} \sum_{t=r_i}^{d_i} x_{idt} = 1$	$\forall i \in I$	(2)
$\sum_{i \in I} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1$	$\forall t \in T, d \in D$	(3)
$\sum_{d \in D} \sum_{t \in T} (t + p_{id} - 1) \cdot x_{idt} - d_o \leq M \cdot y_{io}$	$\forall i \in I, o \in O$	(4)
$x_{idt} \in \{0,1\}$	$\forall i \in I, d \in D, t \in T$	(5)
$y_{io} \in \{0,1\}$	$\forall i \in I, o \in O$	(6)

Source: own table

The TSFD-DT aims to minimize the total number of delayed product units (1). Constraints (2) compel that each inbound truck is processed once and that truck processing starts within a truck's time window. Inequalities (3) assure that at most one inbound truck can be processed at a dock-door at a time. Moreover, constraints (4) determine whether inbound truck i's cargo arrives in the outbound area before the loading operations of outbound truck o start (i.e., $y_{io} = 0$) or not (i.e., $y_{io} = 1$). Lastly, the decision variables are defined in (5) and (6).

A significant drawback of the presented discrete-time model formulation is the time-indexation, which inevitably results in a huge number of decision variables for large problem instances, especially when long planning horizons with many time intervals must be considered. In order to overcome this disadvantage, a continuous-time model formulation is proposed below. Since time is not modeled explicitly in the formulation, it reduces the number of decision variables significantly.

The continuous-time formulation, denoted as TSFD-CT, applies a set of binary decision variables x_{id} for assigning inbound trucks $i \in I$ to dock-doors $d \in D$ and a set of continuous variables s_i to indicate the associated start times of the inbound trucks. An additional set of binary decision variables ϕ_{ij}, defined for every truck pair $(i,j) \in I^2$, is introduced for the sake of determining the truck sequence at a dock-door. ϕ_{ij} signals whether truck i starts before truck j (i.e., $s_i \leq s_j \Rightarrow \phi_{ij} = 1$) or truck j starts before truck i (i.e., $s_j < s_i \Rightarrow \phi_{ij} = 0$). When applying the notation summarized in Table 3, the TSFD-CT can be formulated as shown in Table 4.

Table 3. Additional and altered notations for the continuous-time model formulation of the TSFD

Decision variables:	
x_{id}	Binary decision variable: 1, if inbound truck $i \in I$ is assigned to door $d \in D$; 0, otherwise.
s_i	Continuous decision variable: Start time of inbound truck $i \in I$.
ϕ_{ij}	Binary decision variable: 1, if processing of inbound truck $i \in I$ starts before processing of inbound truck $j \in I$ starts; 0, otherwise.

Source: own table

Table 4. Continuous-time model for the TSFD

Objective function:		
$Minimize \sum_{i\in I}\sum_{o\in O} f_{io} \cdot y_{io}$		(7)
Constraints:		
$\sum_{d\in D} x_{id} = 1$	$\forall i \in I$	(8)
$r_i \leq s_i \leq d_i$	$\forall i \in I$	(9)
$s_i + p_{id} \cdot x_{id} + M \cdot (x_{id} + x_{jd} + \phi_{ij} - 3) \leq s_j$	$\forall i,j \in I: i \neq j, d \in D$	(10)
$\phi_{ij} + \phi_{ji} = 1$	$\forall i,j \in I: i \neq j$	(11)
$\left(s_i + \sum_{d\in D} p_{id} \cdot x_{id}\right) - d_o \leq M \cdot y_{io}$	$\forall i \in I, o \in O$	(12)
$x_{id} \in \{0,1\}$	$\forall i \in I, d \in D$	(13)
$s_i \geq 0$	$\forall i \in I$	(14)
$\phi_{ij} \in \{0,1\}$	$\forall i,j \in I$	(15)
$y_{io} \in \{0,1\}$	$\forall i \in I, o \in O$	(16)

Source: own table

The objective is to minimize the total number of delayed product units (7). Through constraints (8), every inbound truck is assigned to a dock-door, whereas constraints (9) guarantee that truck processing starts within a truck's time window. Inequalities (10) prevent multiple trucks from being processed simultaneously at the same dock-door. Constraints (11) are introduced to compel a well-defined precedence relation for truck pairs. Moreover, inequalities (12) determine whether an inbound truck i's cargo reaches the outbound area before outbound truck o's deadline. Lastly, the decision variables are defined in (13) to (16).

Model comparison

While the TSFD-DT involves $|I| \cdot (|D| \cdot |T| + |O|)$ decision variables and $|I| \cdot (|O| + 1) + |D| \cdot |T|$ constraints, the TSFD-CT contains $|I| \cdot (|I| + |D| + |O| + 1)$ decision variables and $|I| \cdot (|I| - 1) \cdot (|D| + 1) + |I| \cdot (|O| + 3)$ constraints. Table 5 presents exemplary model dimensions for both model formulations of the TSFD.

Table 5. Exemplary model dimensions for different model formulations of the TSFD

Instance dimensions				TSFD-DT		TSFD-CT	
$\|I\|$	$\|D\|$	$\|T\|$	$\|O\|$	# Decision variables	# Constraints	# Decision variables	# Constraints
50	8	20	48	20,200	1,434	3,950	23,200
50	8	20	96	39,400	1,818	3,950	23,200
50	8	20	240	97,000	2,970	3,950	23,200
100	15	40	48	76,000	4,820	15,600	162,700
100	15	40	96	148,000	5,540	15,600	162,700
100	15	40	240	364,000	7,700	15,600	162,700
200	30	60	48	300,000	13,640	58,200	1,246,400
200	30	60	96	588,000	15,080	58,200	1,246,400
200	30	60	240	1,452,000	19,400	58,200	1,246,400
300	50	80	48	744,000	26,700	129,300	4,599,600
300	50	80	96	1,464,000	29,100	129,300	4,599,600
300	50	80	240	3,624,000	36,300	129,300	4,599,600

Source: own table

The examples show that the TSFD-CT deals with a considerably lower number of decision variables than the TSFD-DT. If a fine time granularity is compulsory, the continuous-time model involves up to 96% fewer decision variables than the discrete-time model. However, this reduction comes at the cost of a higher number of constraints. It can be seen from the examples in the table that the TSFD-CT handles up to ca. 170 times more constraints than the TSFD-DT. However, it is uncertain which MIP formulation has a better computational performance when solving problem instances with an off-the-shelf solver such as CPLEX or Gurobi. Therefore, the computational experiment in the following section aims to identify the best performing MIP formulation.

COMPUTATIONAL EXPERIMENTS

This section sets out to analyze the computational performance of the proposed MIP formulations. For this purpose, test instances that consider the time from 8:00 – 16:00 as the planning horizon are generated. Furthermore, inbound truck arrival times are randomly distributed between 08:00 and 14:30, while outbound truck deadlines are randomly chosen between 13:00 and 16:00. By doing so, the outbound truck deadlines likely affect the scheduling of inbound trucks. Similar to Rijal et al. [2019], we assume that every inbound truck supplies between five and seven outbound trucks. Processing times p_{id} are randomly chosen between 30 and 70 minutes. Table 6 shows the additional parameters that are used to generate the test instances.

Table 6. Parameters for the test instance generation

Parameter	Parameter values
$\|I\|$	30, 50, 80
$\|D\|$	5, 7, 9
Time interval length	10 minutes, 5 minutes, 2 minutes
Inbound truck time windows	30-50 minutes, 60-80 minutes

Source: own work

A total of 180 test instances are randomly generated for the experiment. The experiment is conducted on a notebook with an Intel i7-8550 CPU and 16GB RAM. IBM's ILOG CPLEX Optimizer V12.10.0 is used to solve the MIP formulations. The solution time for all runs is limited to 5 minutes per test instance.

Table 7 reports the summarized computational results of the computational experiment. The TSFD-DT finds the optimal solution for all 180 problem instances in less than one minute. Moreover, the results indicate that the time interval length has a strong effect on the problem complexity of the discrete-time model. The solution time grows disproportionally when the time interval length is decreased. It can also be observed that the instances with wider truck time windows are more challenging to solve than instances with shorter time windows. Wider truck time windows increase the size of the discrete-time MIP model and the size of the solution space, which, in turn, results in longer computational times.

The TSFD-CT, on the other hand, solves 116 out of 120 test instances with 30 or 50 inbound trucks within the time limit. However, the continuous-time formulation requires a much longer solution time than the discrete-time formulation. Moreover, it has difficulties with the larger instances that include 80 inbound trucks. It only solves 8 out of 60 test instances with 80 inbound trucks to optimality. Surprisingly, the TSFD-CT cannot even identify a feasible integer solution in 49 out of 60 large test instances.

It can be summarized that the TSFD-DT clearly outperforms the TSFD-CT when seeking optimal solutions with an off-the-shelf solver such as CPLEX or Gurobi.

Table 7. Numerical results for the different MIP formulations of the TSFD

Instances				**TSFD-DT**			**TSFD-CT**			
$\lvert I \rvert$	$\lvert D \rvert$	Time interval length [min]	Time window length [min]	Avg. CPU time [s]	Optimal solution found	Avg. optimality gap	Avg. CPU time [s]	Feasible solution found	Optimal solution found	Avg. optimality gap
30	5	10	30-50	0.09	10/10	0.0%	0.35	10/10	10/10	0.0%
30	5	10	60-80	0.17	10/10	0.0%	1.44	10/10	10/10	0.0%
30	5	5	30-50	0.27	10/10	0.0%	0.66	10/10	10/10	0.0%
30	5	5	60-80	0.45	10/10	0.0%	4.36	10/10	10/10	0.0%
30	5	2	30-50	1.07	10/10	0.0%	1.21	10/10	10/10	0.0%
30	5	2	60-80	1.84	10/10	0.0%	5.65	10/10	10/10	0.0%
50	7	10	30-50	0.34	10/10	0.0%	34.98	10/10	10/10	0.0%
50	7	10	60-80	0.56	10/10	0.0%	58.52	10/10	9/10	0.2%
50	7	5	30-50	0.81	10/10	0.0%	31.20	10/10	10/10	0.0%
50	7	5	60-80	1.27	10/10	0.0%	101.27	10/10	9/10	2.2%
50	7	2	30-50	3.18	10/10	0.0%	69.10	10/10	9/10	0.4%
50	7	2	60-80	4.95	10/10	0.0%	58.85	10/10	9/10	0.2%
80	9	10	30-50	0.92	10/10	0.0%	239.63	4/10	4/10	60.0%
80	9	10	60-80	1.40	10/10	0.0%	300.00	0/10	0/10	100.0%
80	9	5	30-50	2.36	10/10	0.0%	281.72	4/10	2/10	65.4%
80	9	5	60-80	3.89	10/10	0.0%	300.00	0/10	0/10	100.0%
80	9	2	30-50	13.83	10/10	0.0%	298.96	3/10	2/10	70.1%
80	9	2	60-80	19.96	10/10	0.0%	300.00	0/10	0/10	100.0%

Note: For each parameter combination, ten different test instances are solved.
Source: own table

CONCLUSIONS

In this paper, we proposed both a discrete-time and a continuous-time mixed-integer programming formulation for a variant of the truck scheduling problem with fixed outbound departures. While the discrete-time formulation comes with a large number of decision variables, the continuous-time formulation requires a large number of constraints. Both formulation's computational performance was compared in a computational experiment with 180 test instances.

The experiment revealed that the discrete-time model clearly outperforms the continuous-time model in terms of solution time and solution quality. The discrete-time model can be solved to optimality in a reasonable time with a default solver, even for problem instances with 80 inbound trucks and a fine time granularity.

The tests showed that the solution time grows when increasing the number of trucks and decreasing the time interval length. Thus, the proposed discrete-time formulation may struggle to solve large-sized instances with several hundreds of trucks. Future research could develop solution procedures that can solve large instances of the truck scheduling variant.

ACKNOWLEDGMENTS AND FUNDING SOURCE DECLARATION

This research work is supported by the Bosch-Chair of Global Supply Chain Management at the Sino-German School of Postgraduate Studies, Tongji University.

REFERENCES

Bartholdi J.J., Gue K.R., Kang K., 2008. Staging protocols for unit-load crossdocking, in Lahmar, M. (Ed.), Facility logistics. Approaches and solutions to next generation challenges, Auerbach Publications, Boca Raton, FL, 153-171, http://doi.org/10.1201/9781420013719.

Boysen N. Fliedne, M., 2010. Cross dock scheduling: Classification, literature review and research agenda, Omega, 38(6):413-422, http://doi.org/10.1016/j.omega.2009.10.008

Boysen N., Briskorn D., Tschöke M., 2013. Truck scheduling in cross-docking terminals with fixed outbound departures, OR Spectrum, 35(2):479-504, http://doi.org/10.1007/s00291-012-0311-6.

Buijs P., Vis I.F., Carlo H.J., 2014, Synchronization in cross-docking networks: A research classification and framework, European Journal of Operational Research, 239(3):593-608, http://doi.org/10.1016/j.ejor.2014.03.012.

Chmielewski A., Naujoks B., Janas M., Clausen U., 2009. Optimizing the door assignment in LTL-terminals, Transportation Science, 43(2):198–210, http://doi.org/10.1287/trsc.1090.0271.

Forger G., 1995. UPS starts world's premiere cross-docking operation, Modern Material Handling, 36-38.

Gue K.R., 1999. The effects of trailer scheduling on the layout of freight terminals, Transportation Science, 33(4):419-428, http://doi.org/10.1287/trsc.33.4.419.

Ladier A.L. Alpan G. 2016. Cross-docking operations: Current research versus industry practice, Omega, 62:145-162, http://doi.org/10.1016/j.omega.2015.09.006

Lamorgese L., Mannino C., 2019, A non-compact formulation for job-shop scheduling problems in traffic management, Operations Research, 67(6):1586–1609, http://doi.org/10.1287/opre.2018.1837.

Molavi D., Shahmardan A., Sajadieh M.S, 2018. Truck scheduling in a cross docking systems with fixed due dates and shipment sorting, Computers and Industrial Engineering, 117:29-40, http://doi.org/10.1016/j.cie.2018.01.009.

Rahmanzadeh Tootkaleh S., Fatemi Ghomi S.M., Sheikh Sajadieh M, 2016. Cross dock scheduling with fixed outbound trucks departure times under substitution condition, Computers and Industrial Engineering, 92:50-56, http://doi.org/10.1016/j.cie.2015.12.005.

Rijal A., Bijvank M., de Koster R, 2019. Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors, European Journal of Operational Research, 278(3):752-771, http://doi.org/10.1016/j.ejor.2019.04.028.

Serrano C., Delorme X., Dolgui A, 2017. Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans, International Journal of

Production Economics, 194:102-112, http://doi.org/10.1016/j.ijpe.2017.09.008.

Shakeri M., Low M.Y.H., Turner S.J., Lee E.W., 2012. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock, Computers and Operations Research, 39(11):2564-2577, http://doi.org/10.1016/j.cor.2012.01.002.

Stalk G., Evans P., Shulman L.E., 1992. Competing on capabilities: The new rules of corporate strategy, Harvard Business Review, 70(2):57-68.

Tadumadze G., Boysen N., Emde S., Weidinger F., 2019. Integrated truck and workforce scheduling to accelerate the unloading of trucks, European Journal of Operational Research, 278(1):343-362, http://doi.org/10.1016/j.ejor.2019.04.024.

Van Belle J., Valckenaers P., Cattrysse D., 2012. Cross-docking: State of the art, Omega, 40(6):827-846, http://doi.org/10.1016/j.omega.2012.01.005

Van Belle J., Valckenaers P., Vanden Berghe G., Cattrysse D., 2013. A tabu search approach to the truck scheduling problem with multiple docks and time windows, Computers and Industrial Engineering, 66(4):818-826, http://doi.org/10.1016/j.cie.2013.09.024.

Witt C.E., 1998, Crossdocking: Concepts demand choice, Material Handling Engineering, 53(7):44-49.

ROZWIĄZYWANIE PROBLEMU HARMONOGRAMOWANIA PRZEWOZÓW PRZY USTALONYCH ZAŁADUNKACH

STRESZCZENIE. **Wstęp:** Harmonogramowanie przewozów oraz cross-dockingu leży w zasięgu zainteresowania uczonych już od ponad 30 lat. W tym okresie zaproponowało wiele różnych modeli programistycznych tablic awizacyjnych. Jednak zaledwie kilka modeli bierze pod uwagę stałe załadunki, które często są stosowane w przewozach niepełno samochodowych oraz kurierskich. Według naszego rozeznania, żaden z dostępnych modeli nie stosuje modelowania czasem w sposób dyskretny lub ciągły dla uzyskania lepszego wyniku. Celem pracy jest uzupełnienie tej luki w badaniach. Dlatego też rozważono wariant problemu harmonogramowania przewozów ze stałymi załadunkami z celem nadrzędnym znalezienia takiego sposobu harmonogramowania aby minimalizował on liczbę opóźnionych przewozów.

Metody: Zaproponowano dwa modele, opisujące harmonogramowanie przewozów ze stałymi załadunkami. Problem ten został sformułowany poprzez model programistyczny ze zmienną czasu w ujęciu dyskretnym i ciągłym.

Wyniki: Przeprowadzono symulację komputerową w celu określenie działania opracowanych modeli. Porównano wyniki pod względem jakości uzyskanego wyniku oraz niezbędnego czasu dla obliczeń.

Wnioski: Na podstawie uzyskanych wyników można stwierdzić, że proponowany model dyskretny może rozwiązywać problem średniej wielkości w czasie niższej niż minuta. Model oparty na czasie ciągłym uzyskał z kolei optymalizację przy małych przypadkach. Wymagało to jednak dłuższego czasu obliczeniowego. Dodatkowo nie uzyskano dla rozwiązań średniej wielkości czasu niższego od 5 minut. Dlatego też wysunięto wniosek, że model dyskretny jest lepszym w porównaniu z modelem ciągłym.

Słowa kluczowe: cross-docking, harmonogramowanie przewozów, programowanie różnych zmiennych, logistyka, optymalizacja

Pascal Wolff ORCID ID: https://orcid.org/0000-0002-4486-9774
Tongji University, School of Economics and Management
Department of Management Science and Engineering
Siping Road 1500, 200092 Shanghai, **China**
e-mail: wolff@scnm.tu-darmstadt.de

Jiazhen Huo
Tongji University, School of Economics and Management
Department of Management Science and Engineering
Siping Road 1500, 200092 Shanghai, **China**
e-mail: huojiazhen@tongji.edu.cn

Hans-Christian Pfohl
Technical University of Darmstadt, Department of Law and Economics
Supply Chain- and Network Management
Hochschulstraße 1, 64289 Darmstadt, **Germany**
e-mail: pfohl@scnm.tu-darmstadt.de