ISBN: 978-981-18-5852-9
WCSE 2022 Spring Event: 2022 9th International Conference on Industrial Engineering and Applications
doi: 10.18178/wcse.2022.04.098

A Truck Scheduling Problem for Automotive Part Supply with a Point-to-point Network

Ran Lv, Hui Sun +

Department of Industrial Engineering, Southeast University, Nanjing, China

Abstract. This paper studies a scheduling problem based on a point-to-point network, in which automotive parts are delivered with a truck fleet between each single supplier and an original equipment manufacturer (OEM). The objective is to minimize the total transportation cost for the part supply and the inventory cost at the OEM. An integer programming model is proposed to describe the truck scheduling problem. In order to quickly search for good solutions to this NP-hard problem, two heuristic rules and one genetic algorithm are developed. Numerical experiments are conducted to evaluate the performance of the proposed heuristic solution procedures. It shows that the proposed heuristics can solve the scheduling problem effectively and efficiently. Moreover, the genetic algorithm significantly outperforms the two rule-based heuristics.

Keywords: scheduling, automotive part logistics, direct shipping, integer programming, genetic algorithm.

1. Introduction

Nowadays the strategy of mass customization has been extensively applied in the automotive industry. It inevitably results in an increasing product variety that can be observed at many original equipment manufacturers (OEMs). Consequently, just-in-time part logistics becomes one of the imperative challenges in today's automotive production. OEMs need to effectively coordinate thousands of parts and suppliers, a great number of different vehicles and equipment, and hundreds of logistics workers to satisfy the part requirements from the final assembly lines.

The external part logistics involves all activities that supply parts from an external supplier's facility to the OEM's plant. Parts are produced at the respective supplier and shipped to the OEM plant typically by truck. In general, there are three major pathways for transporting automobile parts that are delivered frequently in small lot sizes, namely, point-to-point network, milk-run system, and cross docking system [1].

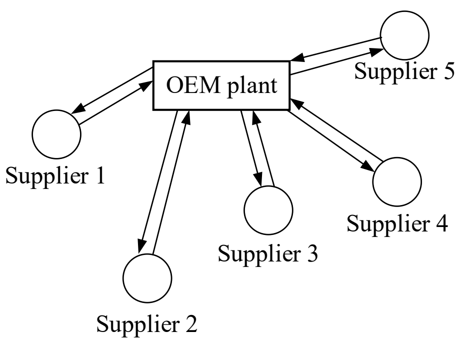

Fig. 1: A point-to-point network consisting of one OEM and five suppliers.

In a point-to-point network (as shown in Figure 1), also referred to as a direct shipping network, each single supplier directly ships its parts to the OEM plant. Direct shipping is especially useful for transporting large and valuable parts with high variety such as car seats. These parts are often sorted by suppliers in the same order just as defined by the production sequence, which can not only reduce inventories but also ease the logistics burden on assembly workers at the OEM [2]. In addition, note that the suppliers in the network

+ Corresponding author. Tel.: +86 25 52090501; fax: +86 25 52090504.
E-mail address: hui.sun@seu.edu.cn

are normally located very close to the OEM plant since frequent small-lot deliveries are needed.

The scheduling and routing of trucks serving a point-to-point network in the context of just-in-time (JIT) production has to consider the trade-off between shipping cost for the part supply and inventory cost at the OEM. This is different from the classical routing problem, in which the primary objective is to minimize the number of vehicles or minimize the total cost related to transportation such as travel distance and time of vehicles [3]. There is abundant literature addressing the vehicle routing problem (VRP). As the classical VRP was proved to be NP-hard [4], researchers developed various heuristic algorithms (e.g., tabu search [5], ant colony optimization [6], particle swarm optimization [7], and genetic algorithm [8]), aiming at finding high quality solutions rapidly. However, very few works discussed the truck scheduling associated with a point-to-point delivery network, which can be considered as a special VRP with time windows.

Point-to-point shipping networks were mainly studied from the viewpoint of strategy or supply chain management [9-11]. Thus, the goal of determining the optimal long-term supply policies was normally adopted. To the best of our knowledge, only two papers studied direct shipping from the viewpoint of scheduling and routing. Emde and Zehtabian [12] developed heuristics to solve the problem of scheduling transportation tasks in a one-supplier/multiple-customer network given a finite planning horizon. Gschwind et al. [13] investigated the scheduling problem in a direct shipping network with multiple suppliers and customers, and proposed an exact branch-cut-and-price algorithm to solve it. In fact, although direct shipping is widely used in many industries, there are still many problems regarding vehicle scheduling and routing that remain untackled.

This paper investigates a new truck scheduling problem for automobile part supply via a point-to-point network consisting of one customer (i.e., OEM plant) and multiple suppliers, which has not been studied explicitly yet. The objective is to minimize the total transportation and inventory costs. It has three contributions as follows.

1) An integer programming model is presented to describe the truck scheduling problem based on a direct shipping network. The model considers not only the conventional constrains regarding task assignment but also the restrictions on parts delivery. The objective is to minimize the total transportation and inventory costs.
2) Heuristic approaches including two simple rules and one genetic algorithm (GA) are presented, aimed at quickly finding good solutions to the truck scheduling problem.
3) Computational experiments are conducted to test the performance of the proposed solution procedures, which are shown to be effective and efficient. Moreover, the GA algorithm significantly outperforms the simple rules.

The rest of this paper is organized as follows: Section 2 gives a detailed description and a mathematical formulation of the truck scheduling problem. In order to solve the problem rapidly, two heuristic rules and one GA are proposed in Section 3. Computational results based on designed problem instances are displayed in Section 4. Finally, Section 5 concludes the paper.

2. Problem Description

Consider an automotive OEM plant that is supplied with N parts by external suppliers via a point-to-point network. The transportation tasks are undertaken by a fleet of K trucks. The truck scheduling problem decides on the allocation of each transportation task, and the schedule of each truck, so that the total transportation and inventory costs are minimized.

The following decision environment is assumed:

- Each part is supplied by a specific supplier and each supplier delivers one part only.
- The consumption rate of each part is constant and known in advance.
- All trucks in the fleet are identical and thus have the same carrying capacity.
- While executing a task, the truck starts from the terminal at the plant, picks up the part in the associated supplier's facility, and returns to the plant.
- Full truckload shipment is used whenever possible in order to save the transportation cost.

- The planning horizon is equally divided into multiple time slots. Trucks can only leave the plant at the beginning of a time slot.
- The transportation cost for a part is only determined by the number of trips to deliver this part.
- The inventory holding cost of each part is known.

Using the notations listed in Table 1, the truck scheduling problem under study can be depicted by the integer programming model as follows.

Minimize

$$C = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} t_i \cdot x_{ijk} + \sum_{i=1}^{N} \sum_{j=1}^{J} h_i \cdot I_{it} \quad (1)$$

Subject to

$$y_{ijk} \leq M \cdot x_{ijk} \qquad \forall i, j, k \quad (2)$$

$$x_{ijk} \leq y_{ijk} \qquad \forall i, j, k \quad (3)$$

$$\sum_{i=1}^{N} x_{ijk} \leq 1 \qquad \forall j, k \quad (4)$$

$$\sum_{i=1}^{N} x_{ij_1k} \cdot T_i - (j_2 - j_1) \leq M \cdot z_{j_1j_2k} \qquad \forall k, j_1, j_2, 1 \leq j_1 < j_2 \leq J \quad (5)$$

$$1 + \sum_{i=1}^{N} x_{ij_2k} - \sum_{i=1}^{N} x_{ij_1k} \leq M \cdot \left(1 - z_{j_1j_2k}\right) \qquad \forall k, j_1, j_2, 1 \leq j_1 < j_2 \leq J \quad (6)$$

$$y_{ijk} \leq Cap_i \qquad \forall i, j, k \quad (7)$$

$$\sum_{k=1}^{K} \sum_{j=1}^{J} y_{ijk} = D_i \qquad \forall i \quad (8)$$

$$\sum_{k=1}^{K} \sum_{j=1}^{J} x_{ijk} = \lceil D_i / Cap_i \rceil \qquad \forall i \quad (9)$$

$$y_{ijk} = z_{ijk}^1 \cdot Cap_i + z_{ijk}^2 \cdot r_i \qquad \forall i, j, k \quad (10)$$

$$z_{ijk}^1 + z_{ijk}^2 \leq 1 \qquad \forall i, j, k \quad (11)$$

$$I_{ij} = I_{i,j-1} + \sum_{k=1}^{K} \sum_{j'=j-1-T_i}^{j-T_i} y_{ij'k} - c_i \qquad \forall i, j = 2, \ldots, J \quad (12)$$

$$I_{ij} \geq c_i \qquad \forall i, j = 1, \ldots, J-1 \quad (13)$$

$$I_{iJ} \geq 0 \qquad \forall i \quad (14)$$

$$x_{ijk} = \{0,1\} \qquad \forall i, j, k \quad (15)$$

$$y_{ijk} \geq 0 \qquad \forall i, j, k \quad (16)$$

$$z_{i_1j_2k} = \{0,1\} \qquad \forall j_1 = 1, \ldots, J, j_2 = 1, \ldots, J, k \quad (17)$$

$$z_{ijk}^1, z_{ijk}^2 = \{0,1\} \qquad \forall i, j, k \quad (18)$$

Objective (1) aims to minimize the total transportation cost for the part supply and inventory cost at the OEM plant. Constraints (2) and (3) jointly ensure that an actual shipping task (i.e., $y_{ijk} > 0$) can be issued only if the associated assignment has been determined (i.e., $x_{ijk} = 1$). Constraints (4) guarantee that each vehicle can be assigned at most one transportation task simultaneously. Constraints (5) and (6) collectively ensure that a transportation task cannot be assigned to a truck until the last task assigned has been finished. Constraints (7) require that the quantity of part (say i) associated with any transportation task cannot exceed the truck loading capacity for this part. Constraints (8) ensure that the total demand for each part in the planning horizon is satisfied. Constraints (9) specify the number of transportation tasks for each part, which determines the transportation cost for part supply. It implies that the first term representing the transportation cost in (1) can be deleted. Constraints (10) and (11) jointly ensure that the decision variable y_{ijk} can only be one of the three values, i.e., 0, Cap_i, r_i, where Cap_i is the truck loading capacity for part i and r_i the quantity of part i that cannot be delivered by a full truckload. Constraints (12) depict the recursive relationship between two adjacent inventories I_{ij} and $I_{i,\,j-1}$. Inequalities (13) and (14) impose the restrictions on the inventory levels for each part. Specifically, at the end of each time slot, the inventory of a part (say i) should be no less than the consumption rate. Moreover, the inventory at the end of the planning horizon should be no less than 0. Constraints (15) and (16) define two decision variables x_{ijk} and y_{ijk}. Finally, three auxiliary binary integer variables are defined in constraints (17) and (18).

As mentioned earlier, the truck scheduling problem under study is a special case of vehicle routing problem, which is claimed to be NP-hard [4]. Thus, in order to deal with the computational intractability and provide real-time good solutions, approximation approaches need to be developed.

Table 1: Notation

Notation	Description
Parameters	
N	Total number of parts (and suppliers), index i
J	Total number of time slots in the planning horizon, index j, j_1, j_2
K	Total number of vehicles, index k
Cap_i	Maximum number of units of part i per truckload
D_i	Total demand of part i in the planning horizon
t_i	Transportation cost per trip for delivering part i
h_i	Inventory holding cost of part i per time slot
c_i	Number of units of part i consumed in one slot
T_i	Number of time slots required per trip to transport part i (to and from supplier i)
M	A very big integer
r_i	Number of units of part i that cannot be delivered by a full truckload, $r_i = D_i - \lfloor D_i / Cap_i \rfloor$
I_{ij}	Inventory level of part i at the end of the jth time slot
Decision variables	
x_{ijk}	Binary decision variable: 1, if a task for transporting part i at the beginning of the jth slot is assigned to vehicle k; 0 otherwise
y_{ijk}	Number of units of part i to be delivered by vehicle k at the beginning of the jth slot
$z_{j_1 j_2 k}$	Binary auxiliary variables
z^1_{ijk}, z^2_{ijk}	Two binary auxiliary variables

3. Heuristic Solution Procedures

Aiming at seeking for good solutions rapidly, this paper presents two simple rule-based heuristics, namely, the fixed route rule and the sequential arrangement rule. Besides, a GA procedure is developed to solve the truck scheduling problem.

3.1. The fixed route rule

Under this rule, each vehicle serves one or multiple suppliers (parts) designated in advance. While in scheduling, a truck, say k for transporting part i, is dispatched once the inventory (of part i) needs to be replenished. When there are multiple requests for different parts simultaneously, the earliest due date rule (EDD) is applied.

3.2. The sequential arrangement rule

According to this rule, the task assignment is made in order of time slots. Within each slot, unassigned tasks are allocated in the order of index numbers of parts to available trucks arbitrarily. This process ends when all the tasks are assigned.

3.3. Genetic algorithm

The GA solution procedure is a biologically inspired metaheuristic and optimization technique. Originally introduced by John Holland in the 1970s, GA has been successfully applied to a wide range of real-world problems of significant complexity. This paper presents a GA to solve the truck scheduling problem based on a point-to-point network. The general procedure of the proposed GA is depicted as follows.

Algorithm 1: Genetic Algorithm
1: Initialize population
2: Check the feasibility of initial solutions
3: **While** iteration < iteration_{max}
4: Calculate the fitness of individuals
5: Select parental population
6: Perform crossover operations
7: Perform mutation operations
8: iteration = iteration + 1
9: **End**
10: **Return** the best solution

1) Chromosomes

A chromosome (individual) in Algorithm 1 corresponds to an actual schedule of all trucks in the fleet. It can be represented by a $K \times J$ matrix S. The element S_{ij} ($k = 1, \ldots, K$; $j = 1, \ldots, J$) in S is determined by

$$S_{kj} = \begin{cases} i, & \text{if truck } k \text{ leaves for supplier } i \text{ at the beginning of the } j\text{th slot} \\ 0, & \text{if truck } k \text{ has no task assigned at the beginning of the } j\text{th slot} \\ -1, & \text{if truck } k \text{ is executing a task at the beginning of the } j\text{th slot} \end{cases} \quad (19)$$

Figure 2 illustrates a schedule of two trucks undertaking 3 tasks for shipping 3 parts (i.e., 1, 2, and 3), respectively. The relationship between S_{kj} and the decision variable for task assignment x_{ijk} is represented by

$$x_{ijk} = \begin{cases} 1, & \text{if } S_{kj} = i \\ 0, & \text{otherwise} \end{cases}. \quad (20)$$

Slot

Truck					
	3	2	−1	1	0
	0	1	3	0	3

Fig. 2: A schedule involving two trucks with five time slots.

2) Fitness function

The fitness function measures the quality of the solutions found by the GA. For the trucking scheduling problem, the objective function value Z is used in the fitness function as represented by

$$fitness = \begin{cases} C, & \text{if solution is feasible} \\ M, & \text{otherwise} \end{cases}, \quad (21)$$

where M is a very large number. Note that since the truck scheduling problem is a minimization problem, a solution with a smaller fitness value is considered a better one.

3) Population initialization

Chromosomes in the population can be generated by repeatedly the run following initialization procedure.

a) Generate a tentative schedule: For each row (say k) of the matrix S, assign an integer v ($0 \le v \le N$) randomly to the element S_{kj} in ascending order of j ($j = 1, \ldots, J$). If v is positive, then the values of the next $T_v - 1$ elements are set to -1, which means the truck is executing a task for delivering part v and not available yet. After all elements are determined, check whether or not the minimum number of deliveries for each part is satisfied. Regenerate S if the check fails.

b) Determine transportation tasks: For each trip in S for part i, randomly determine the actual quantity q (i.e., y_{ijk}) loaded in the truck based on the demand D_i. For extra tasks assigned in S, the corresponding q is set to 0. After all tasks are determined, check whether or not the inventory constraints for each part are satisfied. Return to *a*) if the check fails.

c) Finalize the schedule: Change the values of elements with a corresponding load 0 to 0.

4) Genetic operations

a) Selection: A binary tournament selection strategy is employed. Specifically, for each individual in the population, randomly choose another one and keep the solution with the smaller objective value. In addition, after the tournament selection, an elite strategy is applied. Replace the *m* individuals with the worst fitness value(s) with the *m* individuals in the elite pool.

b) Crossover: Randomly choose two individuals in the population, say *S* and *U*, and then perform a one-point crossover. Specifically, exchange the elements in the first *c* ($1 \leqslant c \leqslant J$) columns of *S* and *U*, where *c* is an integer arbitrarily chosen. Thus, two offspring individuals are produced (as illustrated in Figure 3).

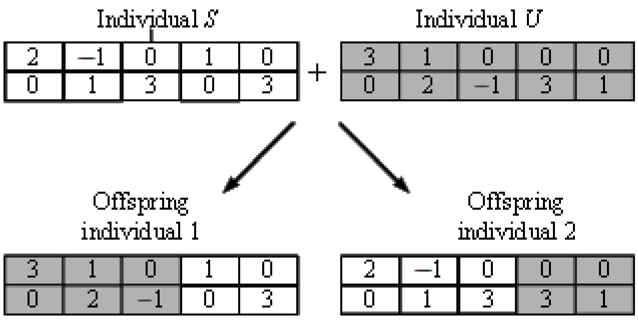

Fig. 3: An illustrative example of crossover operation.

Two repair operations can be performed if the obtained offing individual is infeasible. One of them, namely push operation, delays the assignment of a task until the last one in the schedule has been finished. The task *4* in Figure 4 is delayed to the 5th slot by which the truck delivering part 2 has come back. This operation can be conducted for each truck sequentially.

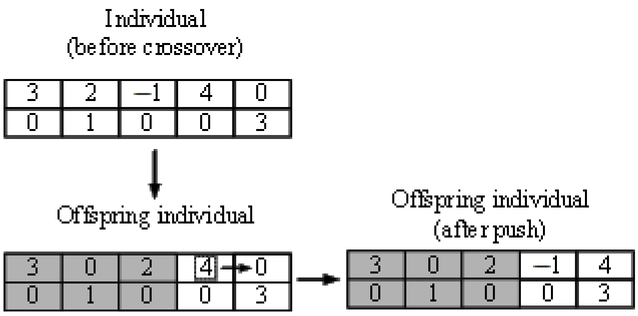

Fig. 4: An illustrative example of push operation.

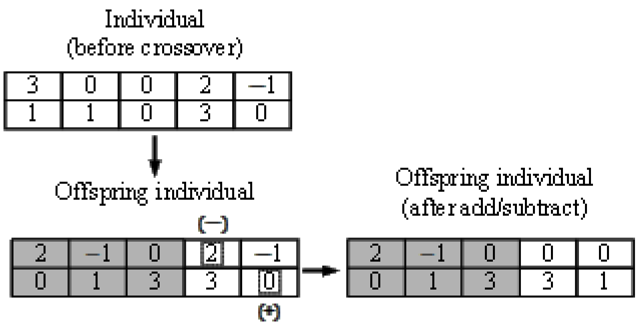

Fig. 5: An illustrative example of add/subtract operation.

The other operation, namely add/subtract operation, removes extra tasks randomly and then adds tasks required to available slots arbitrarily. In Figure 5, the task *2* in the 4th time slot is dropped and a task *1* is added to the 5th time slot. The two operations can be conducted alternately for several times until the resulting schedule is feasible.

c) Mutation: While performing the mutation operation, a new individual is generated by iteratively conducting K swap operations. At each iteration, say $k1$ (k_1 = 1, …, K), an element $S_{k_1 j_1}$ ($1 \le j_1 \le J$) in Matrix S is randomly selected. Then exchange $S_{k_1 j_1}$ with another arbitrarily selected element $S_{k_2 j_2}$ ($1 \le k_2 \le K$, $1 \le j_2 \le J$). In this manner, a new truck schedule can be created. In case it is infeasible, the repair procedure mentioned above is needed.

4. Computational Experiments

Computational experiments are performed to test the performance of the heuristic rules and algorithm presented in this paper. The parameter values used in the experiments are listed in Table 2. Seven instances with different combinations of N, J, and K are generated.

All solution procedures are coded in C++ Language with Microsoft's Visual Studio 2019, and the experiments are run on a 2.40 GHz PC with 16.0 GB RAM. For the GA algorithm, the population size is set to 200, and the number of generations is set to 2000. The crossover rate and mutation rate are set to 0.55 and 0.05, respectively. Besides, the number of chromosomes contained in the elite pool is set to 40. These parameters are determined based on a series of numerical experiments. All instances are solved by using the fixed route rule, the sequential arrangement rule, and the GA, respectively. Thus, three schedules are generated for each instance.

Computation results are displayed in Table 3. For each problem instance, the inventory costs resulting from the three schedules are shown, as the total transportation cost is a constant and thus be discarded. It shows that all the three approaches can solve the truck scheduling problem effectively. Among them, the two rule-based approaches produce comparable results, and the GA obtains the best outcomes for all instances. On average, using the GA to schedule deliveries can save approximately 70% of total inventory costs than applying the two heuristic rules.

Table 2: Parameters used in generating problem instances

Parameter	**Notation**	**Values**
Number of parts	N	3, 4, 5, 8
Number of time slots	J	8, 16
Number of trucks	K	3, 4, 10, 15, 16

Table 3: Computational results

(N, J, K)	**Fixed Route Rule**	**Sequential Arrangement Rule**	**Genetic Algorithm**
(3,8,3)	2840	2680	880
(3,8,3)	4840	4680	1560
(4,8,4)	8600	7880	3520
(5,16,10)	115060	122100	31370
(5,16,15)	131100	149360	28170
(8,16,16)	179740	187340	49530
(8,16,16)	172480	219120	55760

5. Conclusion

In this paper, a new truck scheduling problem associated with supplying automobile parts via a point-to-point network is introduced. An integer programming formulation was proposed to describe the problem. Two heuristic rules and one GA were developed, aimed at finding good solutions rapidly. Computational experiments were conducted to evaluate the performance of the proposed solution procedures. It was shown that all the three approaches can effectively and efficiently generate feasible solutions. Moreover, the GA performs significantly better than the two heuristic rules.

In the future, efforts can be made to look for the optimal combination of the parameters used in GA such as population size, crossover rate, and the number of elites. Besides, other metaheuristics and artificial intelligent approaches (e.g., simulated annealing, particle swarm optimization, deep reinforcement learning, etc.) can be developed. On the other hand, additional restrictions such as time windows of suppliers and variable part consumption rates can be incorporated in the model, and solution procedures can be developed accordingly.

6.References

[1] S. Chopra, and P. Meindl. *Supply Chain Management.* Prentice Hall, 2012.

[2] N. Boysen, S. Emde, M. Hoeck, and M. Kauderer. Part logistics in the automotive industry: Decision problems, literature review and research agenda. *Eur. J. Oper. Res.* 2015, **242**(1): 107-120.

[3] P. Toth, and D. Vigo. *Vehicle routing: problems, methods, and applications.* Society for Industrial and Applied Mathematics, 2014.

[4] J. Lenstra, and A. Kan. Complexity of vehicle routing and scheduling problems. *Networks*. 1981, **11**(2): 221-227.

[5] C. Duhamel, J. Potvin, and J. Rousseau. A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transport. Sci*. 1997, **31**(1): 49-59.

[6] A. Hämmerle, and M. Ankerl. Solving a vehicle routing problem with ant colony optimisation and stochastic ranking. In: R. Moreno Diaz, F. Pichler, and A. Quesada Arencibia (eds.). *Proc. 14th International Conference on Computer Aided Systems Theory – EUROCAST.* Springer. 2013, pp.: 259-266.

[7] B. Moghaddam, R. Ruiz, and S. Sadjadi. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Comput. Ind. Eng.* 2012, **62**(1): 306-317.

[8] A. Kadar, M. Masum, M. Faruque, M. Shahjalal, and H. Sarker. Solving the vehicle routing problem using genetic algorithm. *Int. J. Adv. Comput. Sc*. 2011, **2**(7): 126-131.

[9] G. Gallego, and D. Simchi-Levi. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer R-systems. *Manage. Sci.* 1990, **36**(2): 240-243.

[10] D. Barnes-Schuster, and Y. Bassok. Direct shipping and the dynamic single-depot/multi-retailer inventory system. *Eur. J. Oper. Res.* 1997, **101**(3): 509-518.

[11] S. Chopra. Designing the distribution network in a supply chain. *Transp. Res. E-Log*. 2003, **39**(2): 123-140.

[12] S. Emde, and S. Zehtabian. Scheduling direct deliveries with time windows to minimize truck fleet size and customer waiting times. *Int. J. Prod. Res.* 2019, **57**(5): 1315-1330.

[13] T. Gschwind, S. Irnich, C. Tilk, and S. Emde. Branch-cut-and-price for scheduling deliveries with time windows in a direct shipping network. *J. Scheduling.* 2020, **23**(3): 363-377.