Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 2893–2897

Single Track Transportation in a Two-Machine Production System ⋆

Yakov Zinder * **Alexander Lazarev** ** **Elena Musatova** ***

* *University of Technology, Sydney, Australia (e-mail: Yakov.Zinder@uts.edu.au).*

** *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, Moscow, Russia (e-mail: jobmath@mail.ru)*

*** *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, Moscow, Russia (e-mail: nekolyap@mail.ru)*

Abstract: The paper is concerned with scheduling traffic on a single track between two stations which generate requests for transportation with different release times. These requests are served by a fleet of identical vehicles, each of which can serve (carry) several requests for transportation simultaneously. The single track, used by the vehicles, does not permit traffic in both directions simultaneously. The presented polynomial-time algorithms are based on dynamic programming.

Keywords: Scheduling, dynamic programming, single track, polynomial-time algorithm.

1. INTRODUCTION

The paper is concerned with scheduling traffic between two locations connected by a single track where vehicles cannot travel simultaneously in both directions. The considered scheduling problem arises in flexible manufacturing systems where automated guided vehicles move in both directions between two stations, connected by a fixed guided path made by a wire or marking on the floor Vis (2006). Another application is scheduling automated vehicles in fully automated transportation systems such as the underground automated transportation system of the Amsterdam Schiphol Airport which includes a bidirectional single-track tube van der Heijden et al. (2002). The applications are not limited to automated guided vehicles and include, for example, single track railway in various production systems. Thus, the two considered locations between which it is required to schedule the bidirectional traffic that uses a single track can be workstations, terminals, pickup and delivery points, etc. In this paper the locations will be referred to as stations.

All mentioned above systems are closed ones in the sense that the fleet of vehicles is finite and known. This makes the considered problem related to the intensively studied flow and job shops with transportation Lee and Chen (2001), Nouri et al. (2016). The majority of research in the field of flow and job shops with transportation focuses on simultaneous scheduling the operations on machines and the movement of vehicles. In contrast, in many systems the schedule of requests for transportation is known and fixed, for example, is dictated by the production plan or schedule of arrivals. This paper addresses the latter situation.

There exists a number publications that are concerned with transportation between two locations. For example, a brief survey of publications on two-machine flow shops with transportation can be found in Lee and Chen (2001). Another example is Ilani et al. (2014) that considers transportation between two campuses of an academic college located in two different cities. Such publications, unless being inspired by railway applications, do not consider a single bidirectional track, and therefore allow travelling in both directions simultaneously. As far as railway applications are concerned (see, for example, Brucker et al. (2002), Gafarov et al. (2015)), in contrast to our paper, they do not consider the situation when the same vehicle may visit a location several times, sometimes arriving empty.

This paper is closely related to Zinder et al. (2016), Zinder et al. (2018), Zinder et al. (2020). The transportation problems considered in these three publications can be viewed as problems where a vehicle is instantly available when it is needed and can carry only one request for transportation. In contrast, this paper considers a case when the number of vehicles is limited and each vehicle can carry several requests for transportation which may result in a delay of the departure in order to increase the load and travelling empty.

The considered transportation system is described in Section 2. Section 3 presents an algorithm for scheduling traffic on a single bidirectional track under the restriction that the requests for transportation at each station are numbered in a nonincreasing order of their release times and served in the decreasing order of their numbers. The presented algorithm is suitable for a number of objective functions including the makespan, maximum lateness, total tardiness, total weighted completion time. Section 4 assumes that each request for transportation has an associated deadline. It is shown that the approach, presented in Section 3, allows to solve the problem with such additional restrictions. This section also considers the problem of minimising the average waiting time on

⋆ Acknowledgments: The reported study was partially funded by RFBR, project number 20-38-51010.

10.1016/j.ifacol.2022.10.170

the set of all schedules optimal for the maximum lateness objective function and the problem of minimising the total distance traveled under the restriction imposed by the delivery deadlines.

2. THE TRANSPORTATION SYSTEM

This paper considers a fleet of identical vehicles that serves requests for transportation between two stations, Station 1 and Station 2. Each vehicle can carry up to c requests for transportation simultaneously. The time needed to reach one station from the other is p units of time regardless of the direction and load, including the case when a vehicle is empty. All vehicles use a single track, connecting the stations, that does not permit simultaneous movement in opposite directions.

For $s \in \{1,2\}$, the element, constituting the set $\{1,2\}\backslash\{s\}$, will be denoted by $\bar{s}$. Let N_s be the set of all requests for transportation from Station s to Station $\bar{s}$. Each request for transportation $j \in N_s$ has the associated non-negative release time r_s^j – the time when the corresponding load becomes available for transportation. The first departure can occur at any time $t \geq 0$. Observe that if the first departure occurs at $t = 0$ and all release times are positive, this means that the corresponding vehicle is reallocated without load in order to serve a future request at the opposite station.

At time $t = 0$, there are v_s^0 vehicles at Station s, $s \in \{1,2\}$. So, the fleet of vehicles is comprised of $v_1^0 + v_2^0$ vehicles. In order to avoid a collision, the departure times of any two vehicles, leaving the same station, must differ at least by β time units where $\beta < p$. Since simultaneous movement in both directions is prohibited, the difference between any two departure times from different stations can not be less than p.

A schedule σ specifies for each $s \in \{1,2\}$ and each $j \in N_s$, the arrival time $C_s^j(\sigma)$ of j at Station $\bar{s}$. Since a vehicle can travel from one station to another without any load, the number of runs from Station s to Station $\bar{s}$ in schedule σ, denoted by $a(\sigma, s)$, may exceed $|N_s|$. Therefore, a schedule σ also specifies for each $s \in \{1,2\}$ an increasing sequence

$$S_s^1(\sigma), S_s^2(\sigma), ..., S_s^{a(\sigma,s)}(\sigma)$$

of all departure times from Station s. Consequently, if $j \in N_s$ is delivered as a result of the ith departure from Station s, then

$$C_s^j(\sigma) = S_s^i(\sigma) + p.$$

In what follows, the elements of each N_s are numbered in a nonincreasing order of their release times and are referred to by these numbers. Hence, $N_s = \{1, ..., n_s\}$ where, for any $\{j, g\} \subseteq N_s$, the inequality $j < g$ implies $r_s^j \geq r_s^g$.

3. MAIN ALGORITHM

Assume that the requests for transportation must depart from each station in the decreasing order of their numbers. This is the case, for example, when all release times are different and the arrival times must satisfy the first-in-first-out policy, i.e. for any $s \in \{1,2\}$ and any $\{j, g\} \subseteq N_s$, the inequality $r_s^j < r_s^g$ implies $C_s^j(\sigma) \leq C_s^g(\sigma)$. Consider the following objective function

$$\begin{aligned}\gamma(\sigma) &= \varphi_1^1(C_1^1(\sigma)) \odot \ldots \odot \varphi_1^{n_1}(C_1^{n_1}(\sigma)) \\ &\odot \varphi_2^1(C_2^1(\sigma)) \odot \ldots \odot \varphi_2^{n_2}(C_2^{n_2}(\sigma)) \\ &= \bigodot_{j=1}^{n_1} \varphi_1^j(C_1^j(\sigma)) \odot \bigodot_{j=1}^{n_2} \varphi_2^j(C_2^j(\sigma)),\end{aligned} \tag{1}$$

where, for each $s \in \{1,2\}$ and each $j \in N_s$, $\varphi_s^j(\cdot)$ is a nondecreasing function associated with request j and $\odot$ is a commutative and associative operation such that for any numbers a_1, a_2, b_1, b_2, satisfying $a_1 \leq a_2$ and $b_1 \leq b_2$,

$$a_1 \odot b_1 \leq a_2 \odot b_2. \tag{2}$$

The operation $\odot$ can be, for example, addition or maximum with

$$a \odot b = a + b \quad \text{and} \quad a \odot b = \max\{a, b\}$$

respectively.

Since all φ_s^j are nondecreasing, (2) implies that the objective function γ is nondecreasing. This, in turn, leads to the observation that is stated below as a lemma. Consider an arbitrary schedule σ and a point in time $t \geq 0$. Let $N_s(t)$ be the set of all $j \in N_s$ such that $r_s^j \leq t$ and

$$C_s^j(\sigma) - p \geq t,$$

i.e. $N_s(t)$ is the set of all requests for transportation at Station s that have release times not greater than t and which transportation, according to the schedule σ, commences at or after t. If $N_s(t) \neq \emptyset$, then denote by $x(t)$ the request for transportation with the smallest release time among all requests in $N_s(t)$ and

$$y(t) = x(t) - \min\{|N_s(t)|, c\} + 1.$$

Lemma. *There exists an optimal schedule σ such that, for each departure time $S_s^i(\sigma)$,*

- *if $N_s(S_s^i(\sigma)) = \emptyset$, then the corresponding vehicle leaves Station s without load and its departure time $S_s^i(\sigma)$ is either 0, or $S_s^{i-1}(\sigma) + \beta$, or $C_{\bar{s}}^g(\sigma)$ for some $g \in N_{\bar{s}}$;*
- *if $N_s(S_s^i(\sigma)) \neq \emptyset$, then the corresponding vehicle carries the requests $x(S_s^i(\sigma)), ..., y(S_s^i(\sigma))$ and its departure time $S_s^i(\sigma)$ is either $S_s^{i-1}(\sigma) + \beta$, or $C_{\bar{s}}^g(\sigma)$ for some $g \in N_{\bar{s}}$, or $r_s^{y(S_s^i(\sigma))}$.*

In what follows, only schedules which departure times and workloads of the vehicles satisfy the above lemma will be considered. Furthermore, the only reason for sending a vehicle without load from Station s to Station $\bar{s}$ is the use of this vehicle for transportation from Station $\bar{s}$ to Station s. Therefore, $2(n_1 + n_2)$ can be taken as an upper bound on the number of departures. Consequently, in what follows, without loss of generality, it will be assumed that all possible departure times belong to the following set:

$$\begin{aligned}T = \{t \mid t = m_0 r_s^i + m_1 p + m_2 \beta, \text{ where } s \in \{1,2\}, \\ i \in N_s,\ m_0 \in \{0,1\},\ m_1, m_2 \in \{0,1,\ldots,2(n_1+n_2)\}\}.\end{aligned} \tag{3}$$

A tuple $(t, s, v_1, v_2, k_1, k_2)$ where

$$t \in T, \tag{4}$$

$$s \in \{1,2\}, \tag{5}$$

$$v_s \geq 1,\ v_{\bar{s}} \geq 0,\ v_1 + v_2 = v_1^0 + v_2^0, \quad (6)$$

$$n_1 \geq k_1 \geq 0,\ n_2 \geq k_2 \geq 0,\ k_1 + k_2 \geq 1, \quad (7)$$

$$\text{if either } k_s = 0 \text{ or } t < r_s^{k_s},\ \text{then } k_{\bar{s}} > 0. \quad (8)$$

will be called a *state*. It is easy to see that the entire transportation process, specified by a schedule σ, can be viewed as a sequence of states where t is a departure time from Station s; v_s is the number of vehicles at Station s at t, including the considered departing vehicle; $v_{\bar{s}}$ is the number of all other vehicles (each of these vehicles is either at Station $\bar{s}$ or is on its way to Station $\bar{s}$); and each k_e is the number of all $j \in N_e$ such that

$$C_e^j(\sigma) - p \geq t.$$

If, for a schedule σ, in the corresponding sequence of states, after state $(t, s, v_1, v_2, k_1, k_2)$ the next state is $(t', s', v_1', v_2', k_1', k_2')$, then

$$v_s' = v_s - 1 \quad \text{and} \quad v_{\bar{s}}' = v_{\bar{s}} + 1. \quad (9)$$

As far as t' and k_e' are concerned, it is convenient to introduce the following notations:

$$\mu(t, s, k_s) = \begin{cases} 0, & \text{if either } k_s = 0 \text{ or } t < r_s^{k_s} \\ \chi(t, s, k_s), & \text{otherwise} \end{cases},$$

where

$$\chi(t, s, k_s) = \max\{j : j \leq \min\{k_s, c\};\ t \geq r_s^{k_s - j + 1}\},$$

and

$$h(s, s') = \begin{cases} \beta, & \text{if } s = s' \\ p, & \text{if } s \neq s' \end{cases}.$$

Observe that

$$\min\{|N_s(t)|, c\} = \mu(t, s, k_s),$$

where $N_s(t)$ corresponds to the considered schedule σ. Then, taking into account the above lemma,

$$k_s' = k_s - \mu(t, s, k_s) \quad \text{and} \quad k_{\bar{s}}' = k_{\bar{s}}, \quad (10)$$

$$t' \in \{t + h(s, s')\} \cup \{r_{s'}^j : r_{s'}^j \geq t + h(s, s');\ j \geq k_{s'}' - c + 1\}. \quad (11)$$

Observe also that $\mu(t, s, k_s)$ is computed using only the information provided by the state $(t, s, v_1, v_2, k_1, k_2)$ and does not use any additional information about the corresponding schedule.

As has been discussed above, the only reason for sending a vehicle without load from Station s to Station $\bar{s}$ is the use of this vehicle for transportation from Station $\bar{s}$ to Station s. Since all vehicles are identical, this observation implies that, without loss of generality,

$$\text{if } s' = \bar{s} \text{ and either } k_s = 0 \text{ or } t < r_s^{k_s},\ \text{then } t' \geq r_{s'}^{k_{s'}'}. \quad (12)$$

A state $(t, s, v_1, v_2, k_1, k_2)$ such that

$$0 < k_s \leq c; \quad k_{\bar{s}} = 0; \quad t \geq r_s^1. \quad (13)$$

will be called a *final state*. Given the above lemma, if $(t, s, v_1, v_2, k_1, k_2)$ is the last state in the sequence of states associated with some schedule σ, then $(t, s, v_1, v_2, k_1, k_2)$ is a final state. A state $(t, s, v_1, v_2, k_1, k_2)$ will be called feasible if it is either a final state or there exists a feasible state $(t', s', v_1', v_2', k_1', k_2')$ that together with $(t, s, v_1, v_2, k_1, k_2)$ satisfies (9) - (12). For any feasible state $(t, s, v_1, v_2, k_1, k_2)$, which is not a final state, denote by $\Omega(t, s, v_1, v_2, k_1, k_2)$ the set of all feasible states satisfying (9) - (12).

Each feasible state $(t, s, v_1, v_2, k_1, k_2)$ induces a set of partial schedules each of which specifies how to transport the requests for transportation, constituting the set $\{j : j \in N_1; j \leq k_1\} \cup \{j : j \in N_2; j \leq k_2\}$. In all these schedules the transportation process commences at the point in time t with the departure at this point in time from Station s, assuming that there are v_1 and v_2 vehicles at Station 1 and Station 2 respectively.

Consider all partial schedules σ induced by a feasible state $(t, s, v_1, v_2, k_1, k_2)$ and denote by $f(t, s, v_1, v_2, k_1, k_2)$ the minimal value of

$$\bigodot_{j \in \{1, \ldots, k_1\}} \varphi_1^j(C_1^j(\sigma)) \odot \bigodot_{g \in \{1, \ldots, k_2\}} \varphi_2^g(C_2^g(\sigma)) \quad (14)$$

over all these schedules. Each such partial schedule defines a sequence of feasible states where the first state is $(t, s, v_1, v_2, k_1, k_2)$, the last state in the sequence is a final one, and for any two consecutive states $(t', s', v_1', v_2', k_1', k_2')$ and $(t'', s'', v_1'', v_2'', k_1'', k_2'')$,

$$(t'', s'', v_1'', v_2'', k_1'', k_2'') \in \Omega(t', s', v_1', v_2', k_1', k_2').$$

Then, for the final states,

$$f(t, 1, v_1, v_2, k_1, 0) = \bigodot_{i=1}^{k_1} \varphi_1^i(t + p) \quad (15)$$

and

$$f(t, 2, v_1, v_2, 0, k_2) = \bigodot_{i=1}^{k_2} \varphi_2^i(t + p). \quad (16)$$

For any other feasible state $(t, s, v_1, v_2, k_1, k_2)$, if either $k_s = 0$ or $t < r_s^{k_s}$, then

$$f(t, s, v_1, v_2, k_1, k_2) = \min_{(t', s', v_1', v_2', k_1', k_2') \in \Omega(t, s, v_1, v_2, k_1, k_2)} f(t', s', v_1', v_2', k_1', k_2') \quad (17)$$

and, if $k_s \neq 0$ and $t \geq r_s^{k_s}$, then

$$f(t, s, v_1, v_2, k_1, k_2) = \bigodot_{i=1}^{\mu(t,s,k_s)} \varphi_s^{k_s - i + 1}(t + p) \odot \min_{(t', s', v_1', v_2', k_1', k_2') \in \Omega(t, s, v_1, v_2, k_1, k_2)} f(t', s', v_1', v_2', k_1', k_2'). \quad (18)$$

Let $r_s^0 = 0$ and $k = \min\{n_s, \max\{n_s - c, 0\} + 1\}$. Consider the set of tuples $(t, s, v_1^0, v_2^0, n_1, n_2)$ where $t \in \{r_s^k, \ldots, r_s^{n_s}\} \cup \{0\}$, and denote by H its subset comprised of all feasible states in this set of tuples.Then, the optimal value of the objective function is

$$\gamma^* = \min_{(t, s, v_1^0, v_2^0, n_1, n_2) \in H} f(t, s, v_1^0, v_2^0, n_1, n_2) \quad (19)$$

and can be obtained using (15)–(18) and dynamic programming. The complexity of the corresponding algorithm is $O((n_1 + n_2)^3 n_1 n_2 (v_1^0 + v_2^0) c)$.

Taking into account (3), it is easy to see that each tuple $(t, s, v_1^0, v_2^0, n_1, n_2)$, where $t \in \{r_s^k, \ldots, r_s^{n_s}\} \cup \{0\}$, is a feasible state. Section 4 below considers scheduling when the requests for transportation have deadlines and, therefore, some tuples may not be feasible states and even H can be empty.

4. TRANSPORTATION WITH DEADLINES

This section assumes that each request for transportation $j \in N_s$ has an associated deadline D_s^j, i.e. its arrival time cannot exceed D_s^j. In what follows, it is also assumed that, for any two requests for transportation $j \in N_s$ and $g \in N_s$ from the same station $s \in \{1, 2\}$, the inequality $j < g$ implies $D_s^j \geq D_s^g$. Observe that a schedule that meets all deadlines may not exist.

Such deadlines can arise in bi-criteria problems with ordered objective functions. As an example, consider the problem of minimising

$$\delta(\sigma) = \sum_{s\in\{1,2\}} \sum_{j\in N_s} C_s^j(\sigma) \tag{20}$$

on the set of all schedules which are optimal for

$$L_{max}(\sigma) = \max_{s\in\{1,2\}} \max_{j\in N_s} \{C_s^j(\sigma) - d_s^j\}, \tag{21}$$

where d_s^j is the due date associated with request j, and all these due dates satisfy the condition that, for any two requests for transportation j and g from the same station s, the inequality $r_s^j > r_s^g$ implies $d_s^j \geq d_s^g$.

Indeed, for each station, if when assigning numbers to requests for transportation as specified in Section 2, i.e. in a nondecreasing order of their release times, requests with equal release times receive numbers in a nondecreasing order of their due dates, then for any two $j \in N_s$ and $g \in N_s$, the inequality $j < g$ implies $r_s^j \geq r_s^g$ and $d_s^j \geq d_s^g$. It is easy to see that, among all schedules optimal for (21), there exists a schedule according to which the requests departs from each station in a nonincreasing order of their numbers. Such a schedule can be found, using the algorithm presented in Section 3 where

$$\varphi_s^j(C_s^j(\sigma)) = C_s^j(\sigma) - d_s^j.$$

Let L^* be the optimal value of (21). Then, the bi-criteria problem reduces to the problem of minimising (20) on the set of all schedules σ, satisfying

$$C_s^j(\sigma) \leq L^* + d_s^j \quad \text{for } s \in \{1, 2\} \text{ and all } j \in N_s,$$

where the deadlines $D_s^j = L^* + d_s^j$ satisfy the condition that, for any station s and any $\{j, g\} \subseteq N_s$, the inequality $j < g$ implies $D_s^j \geq D_s^g$. Observe that a desired schedule always exists regardless of the choice of due dates. Furthermore, (20) is (1) with

$$\varphi_s^j(C_s^j(\sigma)) = C_s^j(\sigma),$$

and there exists an optimal schedule for (20) according to which the requests for transportation depart from each station in a nonincreasing order of their numbers.

4.1 Minimisation of γ in the presence of deadlines

Consider the objective function (1) and arbitrary deadlines D_s^j such that, for any two requests for transportation $j \in N_s$ and $g \in N_s$ from the same station $s \in \{1, 2\}$, the inequality $j < g$ implies $D_s^j \geq D_s^g$. As in Section 3, assume that the requests for transportation must depart from each station in the decreasing order of their numbers.

It is easy to see that if there exists a schedule that meets all deadlines, then among these schedules, there exists a schedule according to which all requests for transportation departs from each station in the decreasing order of their numbers. Each such schedule has the property

$$S_s^{i(j,s)}(\sigma) + p \leq D_s^j \text{ for } s \in \{1, 2\} \text{ and all } j \in N_s, \tag{22}$$

where $i(j, s)$ is a number of the vehicle departure with request $j \in N_s$. Therefore, the sequence of states that is associated with such schedule contains only states $(t, s, v_1, v_2, k_1, k_2)$ satisfying the condition

$$\text{if } k_s > 0 \text{ and } t \geq r_s^{k_s}, \text{then } t + p \leq D_s^{k_s}. \tag{23}$$

This observation leads to the following change in the definition of a feasible state originally given in Section 3: a state $(t, s, v_1, v_2, k_1, k_2)$ is feasible if it satisfies (23) and it is either a final state or there exists a feasible state $(t', s', v_1', v_2', k_1', k_2')$ that together with $(t, s, v_1, v_2, k_1, k_2)$ satisfies (9) - (12).

If the set H (see Section 3) is empty, then no feasible schedules exist. If $H \neq \emptyset$, then the optimal value of the objective function is given by (19) and can be obtained using (15)–(18) and dynamic programming.

4.2 Minimisation of the number of departures

Recall that the number of departures from Station s in schedule σ has been denoted by $a(\sigma, s)$. Therefore, the number of departures, or equivalently, the number of vehicle runs in schedule σ is

$$\eta(\sigma) = a(\sigma, 1) + a(\sigma, 2). \tag{24}$$

The objective function (24) differs from (1) and, therefore, the algorithm in Section 3 requires some changes, although, as will be shown below, the definition of a state and the definition of a feasible state remain the same as in Subsection 4.1.

Indeed, consider an arbitrary schedule σ that meets all deadlines and the schedule σ' with the same departure times from the same stations as in σ in which the requests for transportation depart in the decreasing order of their numbers. Since for any requests j and g from the same station s, the inequality $j < g$ implies $D_s^j \geq D_s^g$, the schedule σ' also meets all deadlines. Furthermore, $\eta(\sigma) = \eta(\sigma')$. Therefore, without loss of generality, only schedules where the requests for transportation depart from each station in the decreasing order of their numbers can be considered. Consequently, the definition of a state and the definition of a feasible state remain the same as in Subsection 4.1.

Similar to Section 3, consider all partial schedules σ induced by a feasible state $(t, s, v_1, v_2, k_1, k_2)$ and denote by $z(t, s, v_1, v_2, k_1, k_2)$ the minimal number of vehicle runs among all these partial schedules. For the feasible final states,

$$z(t, 1, v_1, v_2, k_1, 0) = 1, \tag{25}$$

$$z(t, 2, v_1, v_2, 0, k_2) = 1. \tag{26}$$

For all other feasible states $(t, s, v_1, v_2, k_1, k_2)$,

$$z(t, s, v_1, v_2, k_1, k_2) = 1 \tag{27}$$

$$+ \min_{(t',s',v_1',v_2',k_1',k_2') \in \Omega(t,s,v_1,v_2,k_1,k_2)} z(t', s', v_1', v_2', k_1', k_2').$$

If the set H (see Section 3) is empty, then no feasible schedules exist. If $H \neq \emptyset$, then the optimal value of the objective function is

$$\eta^* = \min_{(t,s,v_1^0,v_2^0,n_1,n_2) \in H} z(t, s, v_1^0, v_2^0, n_1, n_2)$$

and can be obtained using dynamic programming.

5. CONCLUSION

The paper presents polynomial-time algorithms for scheduling a homogeneous fleet of vehicles that use a single track between two stations. The directions of further research may include cases with more complex structure of the transportation system, as well as a heterogeneous fleet of vehicles and/or several types of the requests for transportation.

REFERENCES

Brucker, P., Heitmann, S., and Knust, S. (2002). Scheduling railway traffic at a construction site. *OR Spectr.*, 24, 19–30.

Gafarov, E., Dolgui, A., and Lazarev, A. (2015). Two-station single-track railway scheduling problem with trains of equal speed. *Computers & Industrial Engineering*, 85, 260–267.

Ilani, H., Shufan, E., Grinshpoun, T., Belulu, A., and Fainberg, A. (2014). A reduction approach to the two-campus transport problem. *J. Sched.*, 17, 587—599.

Lee, C.Y. and Chen, Z.L. (2001). Machine scheduling with transportation considerations. *J. Sched.*, 4, 3–24.

Nouri, H., Driss, O., and Ghedira, K. (2016). A classification schema for the job shop scheduling problem with transportation resources: State-of-the-art review. *AISC*, 464, 1–11.

van der Heijden, M., Ebben, M., Gademann, N., and van Harten, A. (2002). Scheduling vehicles in automated transportation systems: Algorithms and case study. *OR Spectr.*, 24, 31–58.

Vis, I. (2006). Survey of research in the design and control of automated guided vehicle systems. *Europ. J. of Oper. Res.*, 170, 677–709.

Zinder, Y., Lazarev, A., and Musatova, E. (2020). Rescheduling traffic on a partially blocked segment of railway with a siding. *Automation and Remote Control*, 81, 955–966.

Zinder, Y., Lazarev, A., Musatova, E., and Tarasov, I. (2018). Scheduling the two-way traffic on a single-track railway with a siding. *Automation and Remote Control*, 79, 506–523.

Zinder, Y., Lazarev, A., Musatova, E., Tarasov, I., and Khusnullin, N. (2016). Two-station single track scheduling problem. *IFAC-PapersOnLine*, 49, 231–236.