

**Received:** 26 October, 2022  
**Accepted:** 14 November, 2022  
**Published:** 15 November, 2022

**\*Corresponding author:** Frank Werner, Faculty of Mathematics, Otto von Guericke University, PSF 4120, 39106 Magdeburg, Germany, E-mail: frank.werner@ovgu.de

**Keywords:** Train scheduling; Job shop scheduling

**Copyright License:** © 2022 Frank Werner. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://www.peertechzpublications.com>



## Mini Review

# Solving train scheduling problems as a job shop: A brief review

Frank Werner\*

Faculty of Mathematics, Otto von Guericke University, PSF 4120, 39106 Magdeburg, Germany

## Abstract

An interesting practical problem is the single-track train scheduling problem which can be considered a job shop scheduling problem, namely since the sequence of sections is fixed for a train route, it corresponds to fixed machine routes (technological orders) in a job shop scheduling problem. However, for a train scheduling problem, typically some additional constraints such as blocking, sidings, stations with parallel tracks, deadlocks, train length, or headways, etc. have to be considered. The job shop problem has been well investigated in the literature and belongs to the hardest problems in scheduling theory. In this mini-review, some results in this area are discussed, where the main focus is on results that the author has obtained with his collaborators and Ph.D. students during the last decade.

**MSC classification:** 90 B 35

## Introduction

The single-track train scheduling problem can be formulated as follows. A set of  $n$  trains with fixed routes through particular sections from the origin to the destination has to be scheduled. Trains can be considered jobs and the  $m$  single tracks can be interpreted as machines. A train passing through a section or equivalently, a job being processed on a machine, is called an operation. In particular, operation  $O_{ij}$  denotes the  $j$ th section in the route of the train  $i$  with a given travel time  $p_{ij} > 0$  (or equivalently, the processing time of the  $j$ -th operation of the job  $i$ ). Therefore, it is natural to solve the single-track scheduling problem as a job shop scheduling problem which has widely been investigated in the literature. Note that for a train scheduling problem, we have usually  $n \leq m$  while for machine scheduling problems, one has usually  $n \geq m$ . However, for train scheduling problems, in practice, further conditions such as e.g. parallel tracks at sidings or stations as well as deadlock constraints have to be included. If parallel tracks at sidings or stations are considered, the underlying

basic problem is a flexible job shop problem, where several parallel machines can process certain operations.

Typical optimization criteria for train schedules are based on the arrival times  $C_i$  of the train  $i, i \in \{1, \dots, n\}$ , at the destination (or equivalently the completion time of a job). While in scheduling problems, often the maximum completion time

$$C_{max} = \max_{i=1, \dots, n} \{C_i\}$$

should be minimized, for train schedules, frequent sum criteria are considered, e.g. the sum of the transit times of the trains  $\sum C_i$  or if due dates  $d_i$  are given (i.e., the desired arrival times of the trains at the destination), total tardiness

$$\sum T_i = \sum_{i=1}^n \max\{C_i - d_i, 0\}$$

should be minimized. In contrast to the makespan criterion, these sum criteria consider explicitly the arrival times of all trains. In addition, for train scheduling problems, a release

date  $r_i$  is often given for each train  $i$ , which corresponds to the earliest time when the train can start from the origin.

### Solution approaches

Szpigel [1] was the first who formulated a single-track train scheduling problem as a job shop scheduling problem. He developed a branch and bound algorithm which was tested on instances with 10 trains and 5 single-track sections for the minimization of the weighted transit times of the trains ( $\sum w_i c_i$ ), where  $w_i$  is the weight for the train  $i$ . The first survey on optimization models for train schedules has been given by Cordeau, et al. [2]. Later, a broad overview of different problem structures and solution approaches for the period until 2010 has been given e.g. by Lusby, et al. [3].

Sotskov and Gholami [4] applied the well-known shifting bottleneck procedure from job shop scheduling to a single-track train scheduling problem with minimizing total weighted tardiness  $\sum w_i T_i$ . Due to the unary NP-hardness of the resulting single-machine problems in the shifting bottleneck approach, these problems are heuristically solved in each iteration. The algorithm has been tested on small instances and it turned out that the computational times strongly increase with the problem size.

Gholami, et al. [5] considered the single-track train scheduling problem with given release dates using a mixed graph approach. They used several optimization criteria: the minimization of the makespan ( $C_{max}$ ), the minimization of the sum of the train transit times ( $\sum C_i$ ), and the minimization of total tardiness ( $\sum T_i$ ). Due to the NP-hardness of these problems, several constructive heuristics were developed and compared on instances with up to  $n=12$  trains (jobs) and  $m=12$  single tracks (machines). More precisely, the authors applied an ordinal algorithm (ORD), a maximum processing time (MaxPT) and a minimum processing time algorithm (MinPT). The ordinal algorithm generates a sequence of the operations on different machines in the order they are requested for processing the jobs, while the MaxPT (MinPT) algorithm tends to schedule first the jobs with the largest (smallest) total processing time of the jobs on all machines. Each of these heuristics was applied in three variants, namely using the release times (RT), the completion times (CT) and the due dates (DD) as priorities for the ordering of the jobs on the same machine, giving in total 9 heuristics. It turned out that one of the suggested algorithms, called the Ordinal-SCT algorithm (i.e., the ordinal algorithm with the shortest completion time as priority rule), yielded good results for all three optimization criteria, while the use of a more sophisticated algorithm like the shifting bottleneck procedure used much more time but gave only very slight improvements in the objective function value.

Later, Gholami, et al. [6] extended these investigations with the 9 fast edge-orientation heuristics for job shop scheduling with applications to train to schedule. These 9 heuristics were tested on a set of 118 randomly generated instances with up to 2000 operations per instance, namely up to  $n=20$  and  $m=200$  again for the three objective functions considered in [5]. In addition, results have also been given for the Lawrence instances la01 - la20 with the  $C_{max}$  criterion.

The papers discussed above still did not consider additional constraints compared to the classical job shop problems such as e.g.  $J|r_i \geq 0|C_{max}$  or  $J|r_i > 0|\sum T_i$  in the standard scheduling notation. Lange and Werner [7] considered the train scheduling problem in a given railway network by considering single tracks, sidings and stations with permitted recirculation and the objective to minimize total tardiness. Sidings are locations, where trains can pass other trains and sidings and stations can be represented by a set of parallel tracks. This paper considered also blocking constraints, i.e., a train cannot enter a section, where just another train is traveling on. The authors presented four mixed integer programming formulations based on the parallel-machine approach including fixed routes for all trains and on the machine-unit approach with routing flexibility, where two types of decision variables (precedence variables and order variables) were used. Roughly speaking, the parallel-machine approach is based on a random assignment of a machine from a set of parallel machines, while the machine-unit approach integrates an optimal choice of a machine into the optimization process. These formulations are compared on hard scheduling instances with up to 20 trains (jobs) and 11 sections (machines) in terms of the objective function value, the size of the formulation, and the required computational time.

In the paper [8], Lange and Werner presented a permutation-based neighborhood for the blocking job shop problem with the objective of minimizing total tardiness in performing interchanges of adjacent operations on the same machine. The difficulty arising here is that even after an adjacent pairwise interchange (API) of two operations, the resulting solution is not necessarily feasible for the problem with blocking constraints. The authors developed a repair scheme regaining the feasibility for the blocking job shop problem and by this approach, partially defined solutions are extended to complete feasible solutions. The resulting neighborhood is called TAPI ('tardy adjacent pairwise interchange') and tested within a simulated annealing approach for instances with  $n \in \{10, 15\}$  and  $m=11$ .

These investigations have been extended in [9]. Here in particular feasibility and redundancy (i.e., several permutations of the operations represent the same feasible solution) aspects have been discussed. This paper presented a composite neighborhood using both API moves and left shift moves TJ (i.e., leftward shifts of all operations of a tardy job) as well as an 'advanced repairing technique' for guaranteeing feasibility for the blocking job shop problem. It turned out that a composite neighborhood applying an API move with a probability of 0.9 and a TJ move with a probability of 0.1 worked well. Numerical results with the suggested neighborhood, embedded into a simulated annealing framework, have been given and tested on all Lawrence instances la01 - la40 [10] and 15 train-inspired instances from [7].

In [11], motivated by the train scheduling problem, Lange and Werner investigated several heuristic algorithms for the



job shop scheduling problem with blocking constraints and given release dates. They investigated several representation schemes for a solution and presented a basic repair technique constructing a feasible schedule from any given operation permutation and an advanced repair technique defining a feasible neighboring schedule from an initial permutation and the desired interchange. In applying interchange- and shift-based transition schemes, three neighborhood schemes API, TAPI and TJ are defined and investigated. The preferred neighborhood was embedded into a simulated annealing framework and tested on train-inspired instances [7] as well as benchmark instances given by Lawrence [10]. It was shown by Lange [12] that in the case of arbitrary release dates, all three neighborhoods are not necessarily connected. An interesting remaining open question is whether the presented neighborhood is connected in the case when release dates are not present ( $r_i=0$  for all trains), i.e., whether a globally optimal solution can be reached by a sequence of moves from any starting solution in this neighborhood. By comparing two heuristics (based on the API and TAPI neighborhoods, each combined with the TJ neighborhood) with a mixed integer approach, it turned out that the heuristic algorithms yielded optimal or near-optimal solutions for the small- and large-sized instances, where none of the two heuristics dominates the other one.

Results have been also obtained for special cases of the train scheduling problem. Such special problems are of interest e.g. when there is a bottleneck part in the railway network which needs to be scheduled optimally. Gafarov and Werner [13] presented a dynamic programming algorithm of the complexity  $O(n^5)$  and a heuristic algorithm of the complexity  $O(n^3)$  for the two-machine job shop problem with equal processing times on each machine and the objective function  $\sum C_i$  if recirculation is not allowed (i.e., each job consists of exactly two operations). Numerical results have been presented for 30,000 instances with up to 30 jobs. It turned out that the average relative error of the heuristic was less than 1%. This problem corresponds to a train scheduling problem with three sections, where each train has the same speed within a section. The results presented in [13] settled the open complexity status of this problem and extends existing results for the case of a train scheduling problem with only two sections [14]. An interesting remaining open question is whether an NP-hard job shop scheduling problem with equal processing times on each machine and other objective functions exist or whether they are all polynomially solvable in the case of no precedence constraints between the jobs and no job preemption. As an example of a train scheduling problem with a siding, we mention here only the results from [15] for scheduling the two-way traffic between two stations on a single-track network with a siding. It has been shown that for several objective functions, an optimal schedule can be constructed in polynomial time using dynamic programming.

## Conclusion

In the last years, substantial progress in developing algorithms for job shop and flexible job shop problems has been

reached which can also be used for the exact or approximate solution of train scheduling problems. On the other side, a practical railway network consists of a large number  $m$  of tracks and in comparison to machine scheduling problems, a broad number of additional constraints have to be taken into account. In particular, for security reasons, the avoidance of deadlocks is important. A deadlock occurs when two or more trains are preventing each other from moving forward by each occupying the tracks required by the other. In [16], it has been recently shown that the identification of two-train deadlocks can be done in polynomial time. Such investigations could be included in existing train scheduling algorithms in an appropriate form. For future work, it is also necessary to improve the exact and approximate procedures to be able to solve train scheduling problems including the majority of practically relevant constraints with larger values of  $m$  exactly or at least approximately.

## References

1. Szpigel B. Optimal Train Scheduling on a Single Line Railway. *Operations Research*. 1973; 72: 344–351.
2. Cordeau JF, Toth P, Vigo D. A Survey of Optimization Models on Train Routing and Scheduling. *Transportation Science*. 1998; 32(4):380–404.
3. Lusby RM, Larsen J, Ehrgott M, Ryan D. Railway Track Allocation: Models and Methods, *Operations Research Spectrum*. 2011; 33(4):843–883.
4. Sotskov Y, Gholami O. Shifting Bottleneck Algorithm for Train Scheduling in a Single-Track Railway, *Proceedings of the 14<sup>th</sup> IFAC Symposium on Information Control Problems, Part 1, Bucharest / Romania*. 2012; 87– 92.
5. Gholami O, Sotskov YN, Werner F. Job-Shop Scheduling with Objectives Appropriate for Train Scheduling in a Single-Track Railway, *Proceedings of 2<sup>nd</sup> International Conference on Simulation and Modelling, Technologies and Applications - SIMULTECH 2012, Rome / Italy*, 28 - 31 July 2012: 425–430.
6. Gholami O, Sotskov YN, Werner F. Fast Edge-Orientation Algorithms for Job-Shop Scheduling Problems with Applications to Train Scheduling. *International Journal of Operational Research / Nepal*. 2013; 2:19–32.
7. Lange J, Werner F. Approaches to Modeling Train Scheduling Problems as Job-Shops with Blocking Constraints. *Journal of Scheduling*. 2018; 21:191–207. DOI: 10.1007/s10951-017-0526-0.
8. Lange J, Werner F. A Permutation-Based Neighborhood for the Blocking Job-shop Problem with Total Tardiness Minimization. In Klierer, N. Ehmke JF. Borndorfer, R. (eds.), *Operations Research Proceedings 2017*. Springer, 2018: 581 – 586;
9. Lange J, Werner F. A Permutation-Based Heuristic Method for the Blocking Job Shop Scheduling Problem. *IFAC-PapersOnLine*. 2019; 52:1403–1408. DOI: 10.1016/j.ifacol. 2019.11.395.
10. Lawrence S. Supplement to Resource Constrained Project Scheduling: an Experimental Investigation of Heuristic Scheduling Techniques, CSIA, Carnegie Mellon University, Pittsburgh, PA, 1984.
11. Lange J, Werner F. On Neighborhood Structures and Repair Techniques for Blocking Job Shop Problems. *Algorithms*. 2019; 12(11): 242.
12. Lange J. Solution Techniques for the Blocking Job Shop Problem with Total Tardiness Minimization, Ph.D. thesis, Otto-von-Guericke Universität Magdeburg, 2019.
13. Gafarov E, Werner F. Two-Machine Job-Shop Scheduling with Equal Processing Times on Each Machine; *Mathematics*. 2019; 7(3): 301.



14. Gafarov ER, Dolgui A, Lazarev AA. Two-Station Single-Track Railway Scheduling Problem with Trains of Equal Speed, Computers and Industrial Engineering. 2015; 85: 260–267.
15. Zinder Y, Lazarev A, Musatova A, Tarasov I. Scheduling the Two-Way Traffic on a Single-Track Railway with a Siding. Automation and Remote Control. 2018; 79: 506-523.

16. Dal Sasso V, Lamorgese L, Mannino C, Tancredi A, Ventura P. Easy Cases of Deadlock Detection in Train Scheduling. Operations Research. 2022; 70(4).

### Discover a bigger Impact and Visibility of your article publication with Peertechz Publications

#### Highlights

- ❖ Signatory publisher of ORCID
- ❖ Signatory Publisher of DORA (San Francisco Declaration on Research Assessment)
- ❖ Articles archived in worlds' renowned service providers such as Portico, CNKI, AGRIS, TDNet, Base (Bielefeld University Library), CrossRef, Scilit, J-Gate etc.
- ❖ Journals indexed in ICMJE, SHERPA/ROMEO, Google Scholar etc.
- ❖ OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting)
- ❖ Dedicated Editorial Board for every journal
- ❖ Accurate and rapid peer-review process
- ❖ Increased citations of published articles through promotions
- ❖ Reduced timeline for article publication

Submit your articles and experience a new surge in publication services (<https://www.peertechz.com/submission>).

*Peertechz journals wishes everlasting success in your every endeavours.*