WATT -- EURO Working Group On Automated TimeTabling

27-29 AUG 2013

MISTA 2013 PROCEEDINGS

ORBEL
The Belgian Operational Research Society

EVENTMAP

6th Multidisciplinary International Scheduling Conference: Theory & Applications, Gent, Belgium, 27-29 Aug 2013

MISTA 2013

SOLUTION ALGORITHMS FOR THE TWO-STATION SINGLE TRACK RAILWAY SCHEDULING PROBLEM

Evgeny R. Gafarov · Alexandre Dolgui · Alexander A. Lazarev

1 Introduction

A single track network can be seen as an embryonic portion for any type of railway network topology. In this paper, we consider a special **core** subcase with only two stations. This subcase has practical significance and appears often in private railways, e.g., when a company transports loads between two production centers.

The Single Track Railway Scheduling Problem with two stations (STRSP2) is formulated as follows. Given a single track railway between two stations and a set $N' = N'_1 \bigcup N'_2,\ N'_1 \bigcap N'_2 = \emptyset$ of $n' = |N'|$ trains. Trains from the subset N'_1 go from station 1 to station 2, and trains from the subset N'_2 go in the opposite direction. $|N'_1| = n'_1$ and $|N'_2| = n'_2,\ n'_1 + n'_2 = n'$. The track is divided on Q segments $1, 2, \ldots, Q$. Trains from the set N'_1 traverse segments in an order $1 \to 2 \to \ldots \to Q$ and trains from the set N'_2 in the opposite order $Q \to Q-1 \to \ldots \to 1$. At most only one train can be on any track segment at a time[1]. If a train $j' \in N'_1$ is on a track segment, then no train $i' \in N'_2$ can be on the track and vice versa. For each segment $q,\ q = 1, 2 \ldots, Q$, a traversing time p_q is given, in which a train $j' \in N'$ traverses the segment, i.e., for each segment $q,\ q = 1, 2 \ldots, Q$, all the trains go with the same speed[2]. Let $S_{j'}(\Pi)$ and

E.R. Gafarov
Institute of Control Sciences of the Russian Academy of Sciences, Profsoyuznaya st. 65, 117997 Moscow, Russia
E-mail: axel73@mail.ru

A.Dolgui
Ecole Nationale Superieure des Mines, CNRS UMR6158, LIMOS, F-42023 Saint-Etienne, France
E-mail: dolgui@emse.fr

A.A. Lazarev
Lomonosov Moscow State University, Higher School of Economics (National Research University), Institute of Control Sciences of the Russian Academy of Sciences, Profsoyuznaya st. 65, 117997 Moscow, Russia,
E-mail: jobmath@mail.ru

[1] A segment is circumscribed by two signals: one signal from each side, which will control when a train either can or cannot proceed on that segment. This is a safety precaution. So, there is no opportunity for trains to pass each other somewhere between segments.

[2] This assumption is not far away from practice, since most trains travel at a maximal speed allowed.

$C_{j'}(\Pi)$, $j' \in N'$, be the start and completion times of the train j' in a schedule Π, i.e., $S_{j'}(\Pi)$ is a departure time of the train j' from the departure station and $C_{j'}(\Pi)$ is an arrival time to the destination station. Then in a feasible schedule we have:

- $C_{j'} \geq S_{j'} + \sum_{q=1}^{Q} p_q, \ \forall j' \in N'$;
- for any $i' \in N'_1$ and for any $j' \in N'_2$ we have $C_{i'} \leq S_{j'}$ or $C_{j'} \leq S_{i'}$.

In addition, a due date $d_{j'}$, a weight $w_{j'} > 0$, a release date $r_{j'} \geq 0$ (the earliest possible starting time, i.e., $S_{j'} \geq r_{j'}$) for each train $j' \in N'$ can be given. If $C_{j'}(\Pi) > d_{j'}$, then train j' is tardy and we have $U_{j'}(\Pi) = 1$. If $C_{j'}(\Pi) \leq d_{j'}$, then train j' is on-time and $U_{j'}(\Pi) = 0$. Moreover, let $T_{j'}(\Pi) = \max\{0, C_{j'}(\Pi) - d_{j'}\}$ be the tardiness of train j' and $C_{max}(\Pi) = \max_{j' \in N'}\{C_{j'}(\Pi)\}$ is the makespan for schedule Π. For the STRSP2 with release dates the objective is to find an optimal schedule Π^* that minimizes the makespan C_{max} taking into account release dates. This problem is denoted $STRSP2|r_j|C_{max}$ (similar to the traditional three-field notation $\alpha|\beta|\gamma$ for scheduling problems proposed by Graham et al. [2], where α describes the resource environment, β gives the activity characteristics and further constraints and γ describes the objective function). In this paper, we deal with some extensions of STRSP2 with different objective functions and further constraints. We minimize

- number of late trains $STRSP2||\sum U_j$;
- weighted number of late trains $STRSP2||\sum w_j U_j$;
- total completion time $STRSP2|r_j|\sum C_j$ when release dates are given;
- weighted total completion time $STRSP2||\sum w_j C_j$;
- total tardiness $STRSP2||\sum T_j$.

Similar problems arise on a river canal (inland waterways) with a chain of shipping locks [1]. Although, this problem seems to be similar to STRSP2, there are obvious differences between the problems.

To the best of our knowledge there are no publications for this set of STRSP2 problems, although they can be also easily reformulated as shop scheduling problems with Q machines. A literature review on the single track railway scheduling problem can be found, e.g., in [4].

2 Reduction of STRSP2 to a Single Machine Scheduling Problem

Denote $p_{max} = \max_{q=1,2,\ldots,Q}\{p_q\}$ and $P = \sum_{q=1}^{Q} p_q$.

Lemma 1 *Assume that for a train $j' \in N'_1$ we have $C_{j'} = S_{j'} + P$ and train $i' \in N'_1$ is the next train which passes the track. Then, without violation of feasibility conditions the train i' can be scheduled as follows: $S_{i'} = \max\{r_{i'}, S_{j'} + p_{max}\}$ and $C_{i'} = S_{i'} + P$, i.e., the train i' departs from the time point $\max\{r_{i'}, S_{j'} + p_{max}\}$ and leaves without incurring any idle-time.*

Lemma 2 *For any j' and i' belong to the same subset N'_1 or N'_2, in any feasible schedule, we have $|S_{j'} - S_{i'}| \geq p_{max}$ and $|C_{j'} - C_{i'}| \geq p_{max}$.*

Based on these properties, the following reduction to a single machine scheduling problem is proposed.

Single machine scheduling problem A set $N = N_1 \bigcup N_2,\ N_1 \bigcap N_2 = \emptyset$ of n independent jobs that must be processed on a single machine is given. Job preemption is not allowed. The machine can handle only one job at a time. Processing times of jobs are equal to $p,\ \forall j \in N$. For each job $j \in N$, a due date d_j, a weight $w_j > 0$, a release date $r_j \geq 0$ (i.e., the earliest possible starting time) can be given. A feasible solution is described by a permutation $\pi = (j_1, j_2, \ldots, j_n)$ of the jobs of the set N from which the corresponding schedule can be uniquely determined by starting each job as early as possible. Let $S_{j_k}(\pi),\ C_{j_k}(\pi) = S_{j_k}(\pi) + p$ be the start and completion times of job j_k in the schedule resulting from the sequence π. If $j_k \in N_1$ and $j_{k+1} \in N_2$, then between jobs the machine has to be idle during a setup time $st = st_1$. If $j_k \in N_2$ and $j_{k+1} \in N_1$, then between jobs the machine has to be idle during a setup time $st = st_2$. There is no setup time between processing of jobs from the same subset, i.e., $st = 0$. In a feasible schedule $S_{j_{k+1}} = \max\{r_{j_{k+1}}, C_{j_k} + st\}$ holds. Objective functions are the same like for $STRSP2$. If $C_j(\pi) > d_j$, then job j is tardy and we have $U_j(\pi) = 1$, otherwise $U_j(\pi) = 0$. If $C_j(\pi) \leq d_j$, then job j is on-time. Moreover, let $T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$ be the tardiness of job j and $C_{max}(\pi) = \max_{j \in N}\{C_j(\pi)\}$ is the makespan. We note these scheduling problems according to the traditional three-field notation $\alpha|\beta|\gamma$, e.g., $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$ for the single machine scheduling problem with equal-processing-times, setup times and release dates minimizing makespan.

The problems $STRSP2|-|-$ for the previously mentioned objective functions can be reduced to $1|setup-times, N_1, N_2, p_j = p, -|-$ problems as follows. Subset of trains N'_1 corresponds to the subset of jobs $N_1,\ |N_1| = |N'_1|$, and subset N'_2 of trains to the subset $N_2,\ |N_2| = |N'_2|$, of jobs. Let $q,\ q \in \{1, 2, \ldots, Q\}$ be the index of segment for which $p_q = p_{max}$. Denote $TAIL_{left} = \sum_{l=1}^{q-1} p_l,\ TAIL_{right} = \sum_{l=q+1}^{Q} p_l$. Then, assume $p = p_{max},\ st_1 = 2 \cdot TAIL_{right},\ st_2 = 2 \cdot TAIL_{left}$, if $j \in N_1$, then release date $r_j = r_{j'} + TAIL_{left}$, else $r_j = r_{j'} + TAIL_{right}$. If $j \in N_1$, then due date $d_j = d_{j'} - TAIL_{right}$, else $d_j = d_{j'} - TAIL_{left}$. Weights are the same.

A similar reduction can be made for other problem. Thus, instead of STRSP2 the following $1|setup - times, N_1, N_2, p_j = p, -|-$ problems can be considered:

1. $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$;
2. $1|setup - times, N_1, N_2, p_j = p, r_j \sum C_j$;
3. $1|setup - times, N_1, N_2, p_j = p| \sum w_j C_j$;
4. $1|setup - times, N_1, N_2, p_j = p| \sum T_j$;
5. $1|setup - times, N_1, N_2, p_j = p| \sum U_j$;
6. $1|setup - times, N_1, N_2, p_j = p| \sum w_j U_j$.

Some results in equal-processing-time scheduling are presented in [3].

Definition 1. We call schedules for $1|setup - times, N_1, N_2, p_j = p, -|-$ problems *left-shifted*, if they are determined by starting each job as early as possible. Obviously, for any afore mentioned problem there are optimal schedules which are left-shifted.

Definition 2. Let $\Theta = \{t | t = r_j + x_1 \cdot p + x_2 \cdot st_1 + x_3 \cdot st_2,\ j \in \{1, 2, \ldots, n\},\ x_1, x_2, x_3 \in \{0, 1, 2, \ldots, n\},\ x_2 + x_3 \leq x_1\}$.

Notice that there are at most $O(n^4)$ values in set Θ.

Lemma 3 *In all left-shifted schedules, job starting times belong to Θ.*

3 Algorithms for the Problems with Ordered Subsets N_1 and N_2

Lemma 4 *Problems 1-4 are solvable in $O(n^7)$ or in $O(n^6)$ time.*

All the algorithms are based on the same properties of optimal solutions and use the same search procedure.

Denote the subset $N_1 = \{j_1, j_2, \dots, j_{n_1}\}$ and $N_2 = \{i_1, i_2, \dots, i_{n_2}\}$.

Lemma 5 *For each of the above mentioned problems there is an optimal schedule in which jobs are processed in the following special order:*

- *for the problems* $1|setup{-}times, N_1, N_2, p_j = p, r_j|C_{max}$ *and* $1|setup{-}times, N_1, N_2, p_j = p, r_j|\sum C_j$ *jobs are ordered according to non-decreasing release dates, i.e.,* $r_{j_1} \le r_{j_2} \le \dots \le r_{j_{n_1}}$ *and* $r_{i_1} \le r_{i_2} \le \dots \le r_{i_{n_2}}$*;*
- *for the problem* $1|setup - times, N_1, N_2, p_j = p|\sum w_j C_j$ *jobs in each subset are ordered according to non-increasing weights, i.e.,* $w_{j_1} \ge w_{j_2} \ge \dots \ge w_{j_{n_1}}$ *and* $w_{i_1} \ge w_{i_2} \ge \dots \ge w_{i_{n_2}}$*;*
- *for the problem* $1|setup - times, N_1, N_2, p_j = p|\sum T_j$ *jobs in each subset are ordered according to non-decreasing due dates, i.e.,* $d_{j_1} \le d_{j_2} \le \dots \le d_{j_{n_1}}$ *and* $d_{i_1} \le d_{i_2} \le \dots \le d_{i_{n_2}}$*.*

4 Problems with Partially Ordered Subsets

Lemma 6 *For the problem* $1|setup - times, N_1, N_2, p_j = p|\sum w_j U_j$*, there is an optimal left-shifted schedule, where on-time jobs from the same subset* N_1 *or* N_2 *are ordered according to non-decreasing due dates, i.e.,* $d_{j_1} \le d_{j_2} \le \dots \le d_{j_{n_1}}$ *and* $d_{i_1} \le d_{i_2} \le \dots \le d_{i_{n_2}}$*.*

Lemma 7 *Assume, that the jobs are ordered according to Lemma 6. For the problem* $1|setup{-}times, N_1, N_2, p_j = p|\sum U_j$*, there is an optimal left-shifted schedule and such indexes* x*,* $1 \le x \le n_1$ *and* $y, 1 \le y \le n_2$*, that only jobs* $j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}$ *are on-time and processed according to the order given by Lemma 6.*

So, for the problem $1|setup - times, N_1, N_2, p_j = p|\sum U_j$, we have to choose indexes x and y, such that $x + y \to \max$ and jobs $j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}$ can all be processed on-time at the beginning of a schedule. Thus, we have to take into account at most $(n_1{+}1)\log(n_2{+}1)$ pairs (x, y). For each of the pairs we solve the problem $1|setup{-}times, N_1, N_2, p_j = p|\sum T_j$ with set of jobs $\{j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}\}$ by a modification of Algorithm 1. If $\sum T_j(\pi^*) = 0$, then pair (x, y) is feasible. We can conclude the following (see Lemma 8).

Lemma 8 *The problem* $1|setup{-}times, N_1, N_2, p_j = p|\sum U_j$ *can be solved in* $O(n^7 \log n)$ *time.*

For the problem $1|setup - times, N_1, N_2, p_j = p|\sum w_j U_j$, a dynamic programming polynomial time algorithm is suggested. This algorithm based on the following assumptions. Note jobs in $N = \{H_1, H_2, \dots, H_n\}$, where $w_{H_1} \le w_{H_2} \le \dots \le w_{H_n}$. If $w_{H_k} = w_{H_{k+1}}$, then $d_{H_k} \le d_{H_{k+1}}$. Jobs from N_1 and N_2 are noted and ordered according to Lemma 6. Let $H_n \in N_2$ and $H_n = i_k$. For H_n a position in a schedule is defined by a pair (t, l), where $t \in \Theta$ is the starting time of the job, the index $l = 0, 1, \dots, n_1$ means that on-time jobs from the subset $\{j_1, j_2, \dots, j_l\}$ precede the job H_n in a schedule and on-time jobs from the subset $\{j_{l+1}, j_{l+2}, \dots, j_{n_1}\}$ are scheduled after H_n. A position $(-, n_1 + 1)$ means that the job H_n is late and processed at the end of schedule from time $T \in \Theta$.

Then, for each position (t, l) among $O(n^4)$ possible, we can decompose the initial problem into two independent subproblems:

- with a set of jobs $N_{left} = \{j_1, j_2, \ldots, j_l, i_1, i_2, \ldots, i_{k-1}\}$ which have to be processed in interval $[0, t)$;
- with a set of jobs $N_{right} = \{j_{l+1}, j_{l+2}, \ldots, j_{n_1}, i_{k+1}, i_{k+2}, \ldots, i_{n_2}\}$ which have to be processed in interval $[t + p, T)$;

The running time of the Algorithm is $O(n^{15})$.

5 Conclusion

We suppose that running times of Algorithms can be substantivally reduced after their more detailed analysis. Another question arises as for single machine equal-processing-time scheduling problems without setup-times and precedence relations: "Are there problems with equal processing time of jobs, which are NP-hard?"

Acknowledgements The work was partially supported by RFBR (Russian Foundation for Basic Research): 11-08-01321, 11-08-13121, by CNRS (contract 363745) and Saint-Etienne Métropole, France.

References

1. S.Coeney, F.C.R. Spieksmay and G. V. Berghe (2011), The Lockmaster's Problem, ATMOS 2011, Saarbruecken.
2. R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979). Optimization and approximation in deterministic machine scheduling: a survey. Ann. Discrete Math., 5, 287 –326.
3. S. Kravchenko and F. Werner (2011). Parallel Machine Problems with Equal Processing Times: A Survey. Journal of Scheduling, 14(5), 435 - 444.
4. E.S. de Oliveira (2001). Solving Single Track Railway Scheduling Problem Using Constraint Programming. The University of Leeds. School of Computing. PhD Thesis.