Minimization of maximum lateness for M stations with tree topology

Dmitry Arkhipov[1], **Alexander Lazarev**[2], **Elena Musatova**[3]

[3] *Institute of Control Sciences, Moscow, Russia;* `miptrafter@gmail.com`
[2] *Institute of Control Sciences, Moscow, Russia;* `lazarev@ipu.ru`
[3] *Institute of Control Sciences, Moscow, Russia;* `nekolyap@mail.ru`

Minimization maximum weighted lateness for 2 stations. The following problem of scheduling theory is considered. There is a set of orders (wagons) N. Each order $j \in N$ releases on the station A at the moment r_j. Due date of order j equals $d_j = r_j + \delta$. Each order has its own value $w_j > 0$. Wagons are delivered to the station B by train, which covers the distance between A and B in time p. Each train can be departed after the time α of the previous departure. Our goal is to transport all wagons on the station B. The objective function is

$$\min \max_{j \in N} (w_j L_j) \tag{1}$$

We also formulate the ancillary problem with the objective function:

$$\min C_{\max} | (wL)_{\max} < y \tag{2}$$

Schedule which holds (2) we would call $\Theta(N, y)$.
For each values j and y we can determine the moment $t'_j = r_j + \delta + \frac{y}{w_j} - p$. Order j must depart from the station A in the moment which belongs to the interval $[r_j, t'_j)$. We define a set of orders which must be transported to the station B on the train which number is not exceeding m as S_m. Note that on the first step of algorithm sets $S_1, \ldots, S_{m-1}$ are empty and set S_m is full of orders $1, \ldots, n$.
Property 1. We consider the train m which departs at the moment t^m in the schedule $\Theta(N, y)$.
(i) If at the moment t^m there are more than k orders then we should transport on the train m k jobs with minimal t'_j.
(ii) Train m can depart at the moment t^m holds $t^m \geq \max(r_{km}, r(S_m), t^{m-1} + \alpha)$.
(iii) All orders J_l which holds $t^m + \alpha \geq t'_l$ must be transported on the

trains which numbers are not bigger than m, so $J_l \in S_m$.
(iv) When the train m was departed, all orders from the set S_m must had already depart.

Algorithm 1. On each step of algorithm we try to depart the train m from the station A. Firstly, we choose the moment t^m which holds (ii). Secondly, we choose k orders which would be transported on train m with help of (i). After that we check if (iii) holds. If there exists an order $J_l : r_l > t^m, t^m + \alpha \geq t'_l$, so according to (iii) we have to include this order J_l into sets $S_m, S_{m+1}, \ldots, S_q$, then let us return to checking (ii). If there is no such order J_l, (iii) must hold, except the case when we have $x > k$ released orders from the set S_m at the moment t^m (this orders are .... X^0). We obtain $x - k$ orders from X^0, that have to be transported on the trains with number lower than m. After looking for orders $J_j, t'_j > t^m + \alpha$ in the sets $T_{m-1}, T_{m-2}, \ldots$ until we found $x - k$ orders hold this property (let the last one was founded in the train s). After we obtain three sets of orders: $T_s^{m-1} = T_s \cup \cdots \cup T_{m-1}$, X' - set of jobs which holds $\{j | t'_j > t^m + \alpha, j \in T_s^{m-1}\}$ with minimal x moments t' from all such jobs j, and a set X^0. Let us consider the set $X = (T_s^{m-1} \backslash X') \cup X^0$. Orders from this set must be transported on trains $s, \ldots, m$. When we depart this orders we have to change r_{ki} on $r(X_{(i-s+1)k})$ in the property (ii) because we can depart only orders from X. Property (i) holds because all orders which are not belongs to X and released until this moment holds $t' > t^m + \alpha$. Properties (iii) and (iv) hold, except the cases when one of sets S_i changes, if we face it we should go to the next step - considering the train i. If we don't face problems during the transporting set X we should go to the next step - considering the train $m + 1$. This algorithm terminates if on any step we obtain the set S_i with more than ki orders.

Theorem 1. *Algorithm 1 constructs the schedule* $\Theta(N, y)$ *according to criterion* $C_{\max} | (wL)_{\max} < y$. *If algorithm 1 was terminated, there are no schedule* π *holds* $(wL)_{\max} < y$.

Lemma. If there are exist two schedules $\Theta(N, y_1)$ and $\Theta(N, y_2)$ $(y_1 > y_2)$ constructed with help of the algorithm 1, then for each $i = \{1, \ldots, q\}$ and a pair of sets $S_i(\Theta(N, y_1))$ and $S_i(\Theta(N, y_2))$ holds $S_i(\Theta(N, y_1)) \subseteq S_i(\Theta(N, y_2))$

Algorithm 2. Firstly we construct a schedule in which each train departs as soon as possible. then we consider the order j with maximal $w_j L_j$. Order j transports on the train m. If we want to improve the objective function we must transport order j on the train which number is lower than m, so $j \in S_{m-1}$. On the next step we construct the schedule $\Theta(N, w_j L_j)$.We should repeat this operation until we construct the schedule $\Theta(N, y')$ with the objective function y_0, when schedule $\Theta(N, y')$ doesn't exist. On this step we note that $\Theta(N, y')$ is an optimal schedule with the objective function y_0.

Theorem 2. *Algorithm 2 constructs the schedule π which is optimal according to criterion $(wL)_{\max}$ and has minimal $C_{\max}$ among all schedules with the objective function $(wL)_{\max}$.*

Minimization maximum lateness for 3 stations. There are three stations A, B, C and three sets of orders N^{AB}, N^{AC}, N^{BC}. The order $j \in N^{AC}$ releases at the moment r_j^{AC} on the station A and must be transported to the station C. The due date of this order we define as $d_j^{AC} = r_j^{AC} + \delta^{AB} + \delta^{BC}$. Parameters of other orders defines similarly. Train covers the distance between A and B in time p^{AB} and the distance between B and C in time p^{BC}. There are k wagons in each train. Each train can be departed after time α of the previous departure. Let us suppose that $\delta^{AB} > p^{AB} + \alpha$ and $\delta^{BC} > p^{BC}$. the objective function is

$$\min(\max_{j \in N^{AB \cup BC \cup AC}}(L_j)) \tag{3}$$

We also formulate the ancillary problem with the objective function

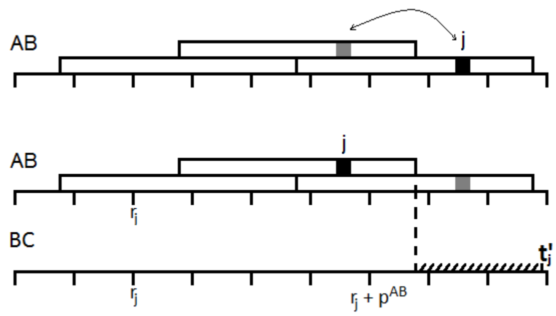
$$\min(C_{\max})|L_{\max} < y \tag{4}$$

The schedule which holds (4) we would call $\Theta^3(N, y)$, $N = N^{AB} \cup N^{AC} \cup N^{BC}$.

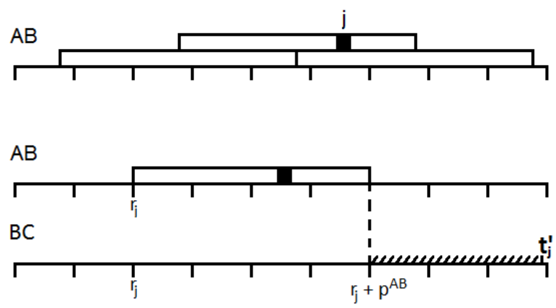
Algorithm 3. Firstly, we construct intervals for each order and each track (AB and BC). The order $j \in N^{AB}$ must be transported on B before the moment $r_j^{AB} + \delta^{AB} + y$, so we obtain that it's interval on the track AB is $[r_j^{AB}, r_j^{AB} + \delta^{AB} + y - p^{AB})$. Order $j \in N^{BC}$, corresponds with the interval $[r_j^{BC}, r_j^{BC} + \delta^{BC} + y - p^{BC})$ on the track BC. The order $j \in N^{AC}$ corresponds with the interval $[r_j^{AC}, r_j^{AC} + \delta^{AB} + \delta^{BC} + y - p^{AB} - p^{BC})$ on the track AB, because this order must be departed from B before the moment $r_j^{AC} + p^{AB}$. We also obtain that order j corresponds with the interval $[r_j^{AC} + p^{AB}, r_j^{AC} + \delta^{AB} + \delta^{BC} + y - p^{BC})$ on the track BC. Each job must start it's transportation on the track in time belongs to the interval which is corresponds with this track. When all interval are constructed we use the algorithm 1 to construct the schedule $\Theta(N^{AB \cup AC}, y)$

corresponds with the track AB and the schedule $\Theta(N^{BC\cup AC}, y)$ corresponds with the track BC, subject to constructed intervals. If one of this schedules isn't constructed successfully our algorithm terminates. If not we pay attention for "bad" orders j which completion time C_j in the schedule $\Theta(N^{AB\cup AC}, y)$ more than it's time of departure in the schedule $\Theta(N^{BC\cup AC}, y)$. If there are no "bad" orders we should depart orders on each track according to it's own schedule Θ. If there are exist some "bad" orders we should find the first of them. We can use two methods to get out of the "bad" order.

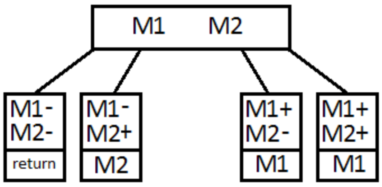
Method 1. We don't change moments of departure of trains on the track AB.



Method 2. We change the moments of departure of trains on the track AB.



We use the following scheme to choose the right method.



If there are no "bad" orders in our pair of schedules $\Theta(N^{AB\cup AC}, y)$ and $\Theta(N^{BC\cup AC}, y)$, then we depart trains on the track AB according to $\Theta(N^{AB\cup AC}, y)$ and on the track BC according to $\Theta(N^{BC\cup AC}, y)$. As a result we obtain the schedule $\Theta^3(N, y)$.

Theorem 3. *Algorithm 3 constructs the schedule $\Theta^3(N, y)$. If algorithm 3 terminates then there is no schedule which holds (4).*

To construct the optimal schedule we use algorithm 2. The only difference is that we should use the schedule $\Theta^3(N, y)$ instead of $\Theta(N, y)$.

M station with tree topology. The formulation of this problem and the problem for 3 stations is the same. The only difference is that we deals with M stations with tree topology. Due to the tree topology there is only one way between each pair of stations. We also can enumerate stations from left to right (or from right to left).

Algorithm 4. We use algorithm 3 to get out of "bad" orders for each station, from the left to the right follows the numeration (according to chosen direction of the train moving). When there are no "bad" orders on each station we obtain the schedule $\Theta^M(N, y)$. After that we use algorithm 2 to construct the optimal schedule for M stations.

Theorem 4. *Algorithm 4 constructs the optimal schedule according to criterion $L_{\max}$ in $O(M^2\frac{n^4}{k})$ operations.*

The authors were supported by the Russian Foundation for Basic Research (project no. 11-08-13121).