# Periodic Metro Scheduling*

Evangelos Bampas, Georgia Kaouri, Michael Lampis, Aris Pagourtzis

School of Elec. and Comp. Engineering, National Technical University of Athens
15780, Athens, Polytechnioupoli, Greece
{ebamp,gkaouri,mlampis,pagour}@cs.ntua.gr

**Abstract.** We introduce the Periodic Metro Scheduling (PMS) problem, which aims in generating a periodic timetable for a given set of routes and a given time period, in such a way that the minimum time distance between any two successive trains that pass from the same point of the network is maximized. This can be particularly useful in cases where trains use the same rail segment quite often, as happens in metropolitan rail networks.
We present exact algorithms for PMS in chain and spider networks, and constant ratio approximation algorithms for ring networks and for a special class of tree networks. Some of our algorithms are based on a reduction to the Path Coloring problem, while others rely on techniques specially designed for the new problem.

**Keywords.** train scheduling, path coloring, delay-tolerant scheduling, periodic timetabling

## 1 Introduction

In railway networks where trains use the same railway segment quite often (e.g. metro) it would be desirable to schedule trains so as to guarantee an ample time distance between successive trains that pass from the same point of the network (in the same direction). Such a scheduling would result in a more delay-tolerant system. This is a particularly essential requirement in cases where there are several overlapping routes that have to be carried out periodically and the time limits are such that some route must start before the termination of another route with which it shares a part of the network.

Here, we formulate this situation by introducing the problem Periodic Metro Scheduling (PMS): given a rail network, a set of routes (described as paths over the network graph), and a time period, we seek to arrange the departure times of routes so that the minimum time distance between any two trains that pass from the same point of the network is maximized. Although

* Research supported in part (a) by the European Social Fund (75%) and the Greek Ministry of Education (25%) through "Pythagoras" grant of the Operational Programme on Education and Initial Vocational Training and (b) by the National Technical University of Athens through "Protagoras" grant.

ATMOS 2006
6th Workshop on Algorithmic Methods and Models for Optimization of Railways
http://drops.dagstuhl.de/opus/volltexte/2006/684

our motivation comes from railway optimization, PMS may also describe other transportation media timetabling problems.

We show the *NP*-hardness of PMS, by reduction from Path Coloring (PC), which is the problem of coloring paths in a graph with minimum number of colors so that intersecting paths receive different colors. We further investigate the relation between the two problems and present exact algorithms for chain and spider networks that rely on a reduction from PMS to PC. Moreover, we show that this technique also applies to rings for which the time needed to traverse the ring is a multiple of the given period. This results to a $\left(\frac{1}{\rho}\frac{L}{L+1}\right)$-approximation algorithm for such instances, where $\rho$ is the approximation ratio we can achieve for PC and $L$ is the maximum number of routes passing through any edge of the network. For ring instances that do not satisfy this condition we present a more involved algorithm that achieves an approximation guarantee of $\frac{1}{6}$. Finally, we show how to apply the path coloring technique to tree networks where the time distance between stations is a multiple of the half of the period, resulting in a $\left(\frac{1}{\rho}\frac{L}{L+1}\right)$-approximation algorithm for this topology as well. Our algorithms employ known algorithms for PC [4, 22, 9, 13] as subroutines.

*Related work.* To the best of our knowledge Periodic Metro Scheduling has not been studied before in the form of an optimization problem. The decision version of PMS, namely the problem of requiring a minimum safety distance not smaller than a given threshold between all pairs of overlapping routes, can be described in terms of a generic problem known as Periodic Event Scheduling Problem (PESP) [23]. PESP has been studied by several researchers, see e.g. [25, 19, 18, 20] and references therein. However, we are not aware of any concrete results for PESP that could apply to PMS, as PESP is usually studied in conjuction with several other constraints that render the problem quite hard and the proposed methods for solving it are mainly heuristics based on "branch-and-bound", "branch-and-cut", and "branch-and-price" methods. A similar problem as PMS has been defined in [11], and it has been proven to be *NP*-complete. However, the setting is broader and the completeness results apply to general graphs[1].

There is a huge bibliography on railway optimization topics; the interested reader is referred to [3] for a nice collection of concepts and earlier results on railway optimization. More recent work on periodic train scheduling includes a rolling stock minimization problem where routes are given and it is sought to determine departure times either arbitrarily (as in our case) or within an allowed time window [7]; however, the objective there is quite different, namely to serve all routes with a minimum number of trains while it is allowed for routes to simultaneously depart from the same station even if they follow the same direction. The rolling stock minimization problem with fixed departure times has been extensively studied: the simplest version, also known as Minimum Fleet Size [2], or Rolling Stock Rostering [8], can be solved exactly in

[1] Unfortunately, we were not able to thoroughly check the similarity of those results to ours because that thesis is available only in German.

polynomial time; Dantzig and Fulkerson [6] give the first known algorithm and Erlebach et al. [8] present one of improved complexity. In [8] some variations are also studied and shown *APX*-hard: allowing empty rides and requiring that the trains pass through a *maintenance station*; constant ratio approximation algorithms have been proposed.

A problem that has recently drawn attention is that of delay management, that is, how to reduce or increase delays of trains in order to better serve railway customers [24, 12, 14]. Several other railway optimization problems have been considered in the literature [26, 5, 21, 1], most of which are hard to solve; a number of heuristics have been presented for them in the corresponding papers.

## 2 Definitions – Preliminaries

We assume that all trains move at the same speed, therefore the duration of traveling between any two connected stations is the same for all trains. In the sequel we denote the travel time between two stations connected by edge $e$ as $t(e)$. We also assume that all edges represent directed railway lines and any two connected stations are linked by a pair of opposite directed edges. For simplicity we consider that the waiting time at stations is negligible.

We are interested in maximizing the time distance between any two overlapping routes, that is, routes that share at least one edge. Note that, due to the uniformity of speed, it suffices to measure the time distance between overlapping routes only at the starting node (station) of the first edge of each common section. More precisely, let $e$ be a common edge between routes $r$ and $r'$ and $t$ (resp. $t'$) be the time at which $r$ (resp. $r'$) enters edge $e$. Then the time distance of $r$ and $r'$ at edge $e$ is defined as $\min(t - t' \mod T, t' - t \mod T)$. When the time distance between two routes in an edge is 0 we say that the routes collide.

We will denote the source of a route $r$ by $s(r)$, and its target by $e(r)$. We define $\tau(i, j)$ as the time distance between nodes $i$ and $j$ in the input graph.

Let us now formally define our problem.

Periodic Metro Scheduling (PMS)
*Input*: A directed graph $G = (V, E)$, an inter-station time function $t : E \rightarrow \mathbb{N}$, an integer time period $T$ and a collection $R = \{r_1, \ldots, r_k\}$ of simple paths on $G$ (routes).
*Feasible solution*: A schedule for $R$, that is, a function $stime : R \rightarrow [0, T)$ which assigns a departure time to each route such that no two routes enter the same edge at the same time.
*Goal*: Maximize the minimum time distance between any two overlapping routes.

We define $L(e)$ to be the number of paths that pass through an edge $e$ of the graph. Let $L = \max_e L(e)$. It is not hard to see that $\frac{T}{L}$ is an upper bound to the objective value of an optimal solution (OPT), because routes cannot be spaced further apart than $\frac{T}{L}$ on the edge with the maximum load.

In our study we will show a close relation between PMS and PC. The definition of PC is as follows:

Path Coloring (PC)
*Input*: A directed graph $G = (V, E)$ and a collection $\mathcal{P} = \{r_1, \ldots, r_k\}$ of paths on $G$.
*Feasible solution*: An assignment of colors to all paths of $\mathcal{P}$ such that no two paths which share an edge are assigned the same color.
*Goal*: Minimize the number of colors used.

PC (note that here we consider the directed version) can be solved optimally in polynomial time in chains (folklore, see e.g. [15]), stars and spiders using $L$ colors, but is known to be *NP*-hard in rings [13] and trees [22].

We will widely use the notation $a \equiv_T b$ to denote that $a \bmod T = b \bmod T$.

## 3 PMS in chain, star and spider networks

A chain is a graph that consists of a single path. A star is a tree with at most one internal node. A spider is a tree in which at most one internal node has degree $\geq 3$, called the central node; that is, a spider is a graph resulting from a star whose edges have been replaced by chains (also called legs of the spider).

In chains we label the nodes of the graph from 0 to $n - 1$ successively. In stars we label the central node 0 and the peripheral nodes $1, \ldots, n$. In a spider with $k$ legs each node is labeled $(i, j)$ where $i = 1, \ldots, k$ represents the leg the node belongs to and $j$ represents the node's position in the leg relative to the central node.

### 3.1 An algorithm for chains

Since all connections are bidirectional we can divide any problem instance into two subproblems, one containing paths moving to the right and one containing paths moving to the left and solve them separately.

Let $t_i$ be the time distance from node $i$ to $i + 1$ for $i = 1, \ldots, n - 1$. In the case of chain networks the time distance between two nodes $i$ and $j$ is

$$\tau(i, j) = \sum_{k=i}^{j-1} t_k$$

We will make use of the fact that PC can be solved optimally for chain networks by using a simple greedy technique.

We propose the following algorithm:

**Algorithm 1** An algorithm for PMS in chain networks

```
1: Compute a path coloring of routes with exactly $L$ colors from $\{0,\dots,L-1\}$. Let $color(r)$ denote the color assigned to route $r$.
2: Set $t = \frac{T}{L}$ and define $L$ time slots as follows: $0, t, 2t, \dots, (L-1)t$.
3: Assign time slots to routes according to the coloring obtained in step 1, namely $timeslot(r) := color(r) \cdot t$.
4: For each $r$ set starting time
       $stime(r) = (timeslot(r) + \tau(0, s(r))) \bmod T$
```

**Theorem 1.** *Algorithm 1 computes an optimal solution for* PMS *in chains.*

*Proof.* Let $r$ and $r'$ be two overlapping paths and without loss of generality assume that $s(r') \leq s(r)$. The first point of their common section is $s(r)$ and their time distance at $s(r)$ is:

$$\begin{aligned} d(r, r', s(r)) = \min\ (&(stime(r') + \tau(s(r'), s(r)) - stime(r)) \bmod T, \\ &(stime(r) - (stime(r') + \tau(s(r'), s(r)))) \bmod T) \end{aligned}$$

Note that

$$\begin{aligned} stime(r') + \tau(s(r'), s(r)) - stime(r) &= timeslot(r') + \tau(0, s(r')) + \tau(s(r'), s(r)) \\ &\quad -(timeslot(r) + \tau(0, s(r))) \\ &= timeslot(r') - timeslot(r) \end{aligned}$$

Therefore

$$\begin{aligned} d(r, r', s(r)) &= \min\left((timeslot(r') - timeslot(r)) \bmod T, (timeslot(r) - timeslot(r')) \bmod T\right) \\ &\geq \frac{T}{L} \end{aligned}$$

Hence, the solution returned by the algorithm is optimal, since $\frac{T}{L}$ is an upper bound for the value of any feasible solution.

□

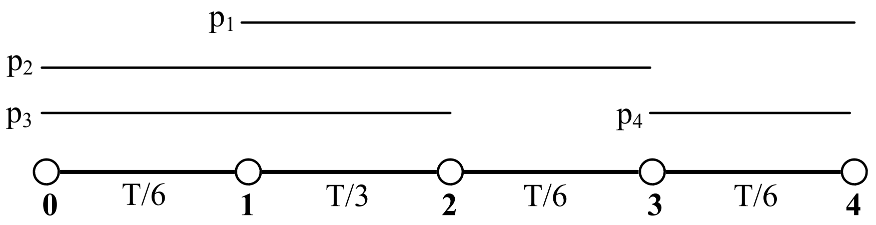

**Fig. 1.** An instance of PMS in chain networks.

*Example 1.* Consider the instance of Figure 1. The maximum load is $L = 3$ and as a result the path coloring algorithm will yield a solution with 3 colors. The time slots corresponding to these colors are: 0, $\frac{T}{3}$ and $\frac{2T}{3}$. Assume that paths $p_3$ and $p_4$ are assigned time slot 0, $p_2$ is assigned time slot $\frac{T}{3}$ and $p_1$ is assigned time slot $\frac{2T}{3}$. According to Algorithm 1, $\mathit{stime}(p_1) = \frac{5T}{6}$, $\mathit{stime}(p_2) = \frac{T}{3}$, $\mathit{stime}(p_3) = 0$ and $\mathit{stime}(p_4) = \frac{2T}{3}$. Observe that on edge $(1, 2)$ the three overlapping paths $p_1$, $p_2$, $p_3$ have time distance at least $\frac{T}{3}$, which is optimal. Furthermore, path $p_1$ reaches node 3 at time $\frac{T}{3}$ ("wrapping around" the end of the time period), thus also having distance $\frac{T}{3}$ from $p_4$.

### 3.2 An algorithm for stars and spiders

Given an instance of PMS in a star or a spider, we will utilize an optimal path coloring of the given instance in order to produce an optimal time schedule; note that such an exact algorithm for spiders can be obtained by appropriate combination of an exact algorithm for stars and the algorithm for chains. We should note that some routes may be confined in one of the spider's legs while others may be directed from one leg to another.

**Algorithm 2** An algorithm for PMS in spider networks

```
1: Compute a path coloring of routes with exactly $L$ colors from $\{0, \ldots, L-1\}$. Let
   $\mathit{color}(r)$ denote the color assigned to route $r$.
2: Set $t = \frac{T}{L}$ and define $L$ time slots as follows: $0, t, 2t, \ldots, (L-1)t$.
3: Assign time slots to routes according to the coloring obtained in step 1, namely
   $\mathit{timeslot}(r) := \mathit{color}(r) \cdot t$.
4: For each $r$ passing through the central node, set starting time
       $\mathit{stime}(r) = (\mathit{timeslot}(r) - \tau(0, s(r))) \bmod T$
5: For each $r$ confined in a single leg and directed towards the central node, set starting
   time
       $\mathit{stime}(r) = (\mathit{timeslot}(r) - \tau(0, s(r))) \bmod T$
6: For each $r$ confined in a single leg and directed away from the central node, set
   starting time
       $\mathit{stime}(r) = (\mathit{timeslot}(r) + \tau(0, s(r))) \bmod T$
```

**Theorem 2.** *Algorithm 2 computes an optimal solution for* PMS *in spiders.*

*Proof.* We will first prove the claim for the case where the spider is a star. Let $r$ and $r'$ be two overlapping routes. Therefore they receive different colors, hence also different time slots. There are two cases: either $s(r) = s(r')$ or $e(r) = e(r')$. In both cases, it suffices to examine their time distance at the central node. Each route arrives at or departs from the central node at time equal to its time slot.

Therefore their time distance is a nonzero multiple of $t = \frac{T}{L}$, which is an upper bound for OPT.

In a general spider network, we consider two cases. For two overlapping routes that pass through the central node, we can use the same argumentation as above for star networks. For two overlapping routes that lie in the same leg, the proof is similar to the proof of Theorem 1 for chains since it can be shown that the same properties hold considering either the central node or the tip of a leg as the first node of the chain (possibly with an appropriate time shift).

□

## 4 PMS in ring networks

In the case of ring networks, that is, networks which consist of a single cycle, we can assume that all trains travel in the same direction (clockwise, without loss of generality), for the same reasons as for chains. Nodes are labeled by picking one arbitrarily and labeling it 0, then labeling every other node $1, \ldots, n-1$ starting from 0's neighbor in the direction trains travel. We define $\tau(i, j)$ as the time distance from node $i$ to node $j$ in the clockwise direction. We also define the ring perimeter $C$ as the total time needed to travel around the ring.

For ring networks we can distinguish between two cases: the case where the ring perimeter $C$ is a multiple of the period $T$ and the case where it is not. In the following two sections we will analyze these cases.

### 4.1 The case $C \equiv_T 0$

**Theorem 3.** *An instance of* PMS *in a ring with $C \equiv_T 0$ admits a solution of value at least $\frac{T}{k}$ if and only if the corresponding* PC *instance can be colored with $k$ colors.*

*Proof.* For the "if" direction, we can produce the desired schedule by using Algorithm 1 for PMS in chains, starting from step 2 and using $k$ instead of $L$. Let $r$ and $r'$ be two overlapping paths; without loss of generality assume that $s(r')$ is closer to 0 than $s(r)$ in the clockwise direction. Because $C \equiv_T 0$, it can be shown that it suffices to check their time distance on only one of their common segments, even if there are two such segments.

Following similar arguments as those in the proof of Theorem 1, it can be shown that the time distance is:

$$\min((\mathit{timeslot}(r) - \mathit{timeslot}(r')) \bmod T, (\mathit{timeslot}(r') - \mathit{timeslot}(r)) \bmod T) \geq \frac{T}{k}$$

For the "only if" direction, suppose we have a schedule for the PMS instance with value $\frac{T}{k}$. We will show how to obtain a coloring with $k$ colors for the corresponding PC instance. For each route $r$, let $\mathit{timeslot}(r) = (\mathit{stime}(r) - \tau(0, s(r))) \bmod T$. Assign to $r$ the color $w - 1$, where $w$ is the smallest integer such that $\mathit{timeslot}(r) < w \cdot \frac{T}{k}$. Since $w$ ranges from 1 to $k$ and for any two overlapping paths $r$ and $r'$ we have $|\mathit{timeslot}(r) - \mathit{timeslot}(r')| \geq \frac{T}{k}$, this is a valid coloring.

□

**Corollary 1.** PMS *in rings is NP-hard.*

*Proof.* We present a reduction from the decision version of PC in rings to the decision version of PMS in rings. PC is known to be *NP*-hard in rings [13]. Suppose we are given an instance of PC in a ring with $n$ nodes and a path set $\mathcal{P}$, asking if $\mathcal{P}$ is colorable with $k$ colors. We construct an instance of PMS in a ring with $n$ nodes, routes identical to the paths in $\mathcal{P}$, inter-station distances of one time unit and $T = n$, asking if it is possible to achieve an objective function value of $\frac{T}{k}$. Clearly, the corresponding PC instance for the PMS instance we produced is the original PC instance. Therefore Theorem 3 applies, implying that the original PC instance can be colored with $k$ colors if and only if a solution of value $\frac{T}{k}$ can be achieved for the PMS instance.

□

At a first glance Theorem 3 seems to imply that a $\rho$-approximation algorithm for PC would give a $\frac{1}{\rho}$-approximation algorithm for PMS. However, this is true only in the case that the optimal solution for the PMS instance divides exactly $T$. In the general case we can show something slightly weaker.

**Theorem 4.** *A $\rho$-approximation algorithm for* PC *in rings implies an* $\left(\frac{1}{\rho}\frac{L}{L+1}\right)$-*approximation algorithm for* PMS *in rings with* $C \equiv_T 0$.

*Proof.* We will use the algorithm of Theorem 3. Let $\mathrm{OPT}_{\mathrm{PMS}}$ be the value of an optimal solution of an instance of PMS and $\mathrm{OPT}_{\mathrm{PC}}$ the cost of an optimal solution of the corresponding PC instance. We observe that $\mathrm{OPT}_{\mathrm{PMS}} < \frac{T}{\mathrm{OPT}_{\mathrm{PC}}-1}$ because a solution of PMS of value $\frac{T}{\mathrm{OPT}_{\mathrm{PC}}-1}$ would lead to a coloring with only $\mathrm{OPT}_{\mathrm{PC}} - 1$ colors by Theorem 3. Recall also that $\mathrm{OPT}_{\mathrm{PMS}} \leq \frac{T}{L}$.

A $\rho$-approximation algorithm for PC returns a solution $\mathrm{SOL}_{\mathrm{PC}} \leq \rho \cdot \mathrm{OPT}_{\mathrm{PC}}$. By Theorem 3 we can compute a solution for PMS of value $\mathrm{SOL}_{\mathrm{PMS}} = \frac{T}{\mathrm{SOL}_{\mathrm{PC}}} \geq \frac{1}{\rho} \cdot \frac{T}{\mathrm{OPT}_{\mathrm{PC}}}$. By the observations above it turns out that:

$$\mathrm{SOL}_{\mathrm{PMS}} \geq \frac{1}{\rho} \cdot \frac{T}{\frac{T}{\mathrm{OPT}_{\mathrm{PMS}}} + 1} = \frac{1}{\rho} \cdot \frac{T \cdot \mathrm{OPT}_{\mathrm{PMS}}}{T + \mathrm{OPT}_{\mathrm{PMS}}} \geq \frac{1}{\rho} \cdot \frac{L}{L+1} \cdot \mathrm{OPT}_{\mathrm{PMS}}$$

□

**Corollary 2.** *There is a* $\left(\frac{2}{3} \cdot \frac{L}{L+1}\right)$-*approximation algorithm and a* $\left(0.73 \cdot \frac{L}{L+1}\right)$-*approximation randomized algorithm for* PMS *in rings with* $C \equiv_T 0$.

*Proof.* By using Theorem 4 and the deterministic approximation algorithm of Karapetian [16] and the randomized approximation algorithm of Kumar [17] that achieve ratios $\frac{3}{2}$ and 1.368 respectively.

□

*Remark 1.* So far, we have assumed that the departure times of trains could be any rational number in $[0, T)$. However there is a possibility that trains need to be assigned integer departure times. In this case following by appropriate modification of the above analysis, it can be shown that our results carry over with a further reduction of the approximation ratio to:

$$\frac{1}{\rho} \cdot \frac{L}{L+1} - \frac{1}{\mathrm{OPT}_{\mathrm{PMS}}}$$

It should be noted though, that this approximation ratio is asymptotically equal to the approximation ratio obtained for rational departure times.

### 4.2 The case $C \not\equiv_T 0$

Consider a ring network with $n$ nodes and two paths $p_1, p_2$ with $0 = s(p_1) < e(p_2) < s(p_2) < e(p_1)$ and $\tau(s(p_1), s(p_2)) = x$. Let $t_1, t_2$ be the moments in time where the trains traveling along $p_1$ and $p_2$ arrive at node 0. These trains reach node $s(p_2)$ at times $(t_1 + x) \bmod T$ and $(t_2 - D + x) \bmod T$ respectively, where $D = C \bmod T$. As a result in order to maximize the minimum distance of the two trains, we have to take into account the following time differences: $(t_1 - t_2) \bmod T$, $(t_2 - t_1) \bmod T$, $(t_1 - t_2 + D) \bmod T$ and $(t_2 - t_1 - D) \bmod T$. It is now clear that the algorithm of Theorem 3 may produce an infeasible solution if $D = (t_2 - t_1) \bmod T$ (see Figure 2). Therefore we need a new algorithm for this case.

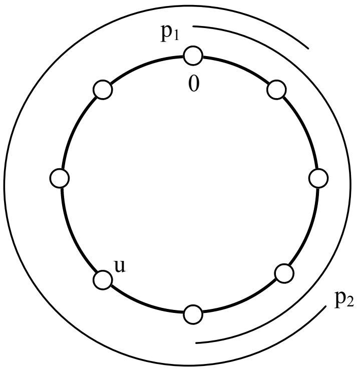

**Fig. 2.** An example showing that the "path coloring" technique does not work for rings with $C \not\equiv_T 0$. Assuming $\tau(0, u) = T$ and $\tau(u, 0) = \frac{T}{2}$, the path coloring technique would assign time slots 0 and $\frac{T}{2}$ to paths $p_1$ and $p_2$ respectively and the two paths would collide at node 0 at any time which is an integer multiple of $T$.

**Theorem 5.** *Let $S$ be a set of paths passing through node $0$ in a ring network. There is an enumeration of $S$ such that for any two paths $p$ and $p'$, if $e(p) > s(p')$ then $p$ appears before $p'$ in the enumeration.*

*Proof.* We define the binary relation $\prec$ over $S$: $p \prec p'$ if and only if $e(p) > s(p')$. This relation is antisymmetric: suppose $p \prec p'$ and $p' \prec p$. Then $e(p') > s(p) > e(p) > s(p')$ which is a contradiction, because for each path $r \in S$ it holds that $s(r) > e(r)$. It is also transitive: if $p \prec r$ and $r \prec q$ then we have $e(p) > s(r) > e(r) > s(q)$, thus $p \prec q$. As a result, the relation $\prec$ is a strict partial order. The theorem follows.

□

**Algorithm 3** An algorithm for PMS in ring networks with $C \not\equiv_T 0$

```
Split $R$ into two sets $\mathcal{P}_0$ and $\mathcal{P}_c$. $\mathcal{P}_0$ contains the paths passing through node 0 (i.e.
    having node 0 as an intermediate node) and $\mathcal{P}_c = R \setminus \mathcal{P}_0$. Let $L_0 = |\mathcal{P}_0|$ and $L_c$ be
    the maximum load with respect to $\mathcal{P}_c$.
Define $t = \frac{T}{6L'}$ and the set of available time slots as follows: $S = \{0, t, 2t, \ldots, (6L' - 1)t\}$, where $L' = \max\{L_0, L_c\}$.
Assign colors to routes of $\mathcal{P}_c$ by using an algorithm for PC in chains.
for each color $k$, $1 \leq k \leq L_c$ do
    define $timeslot(k) = kt$
    for each path $p$ colored with $k$ do
        assign departure time $stime(p) = timeslot(k) + \tau(0, s(p))$.
    end for
    Remove $kt$ from $S$
    Remove from $S$ all time slots whose distance from $kt + D$ is smaller than $t$.
end for
Enumerate paths in $\mathcal{P}_0$ as implied by Theorem 5.
for each $p \in \mathcal{P}_0$ in the order of the enumeration do
    Set $timeslot(p) = wt$, where $wt$ is the smallest available time slot.
    Set $stime(p) = (wt - \tau(s(p), 0)) \bmod T$.
    Remove $wt$ from $S$
    Remove from $S$ all time slots whose distance from $wt + D$ is smaller than $t$.
end for
```

**Theorem 6.** *Algorithm 3 is a $\frac{1}{6}$-approximation algorithm for* PMS *in rings.*

*Proof.* First let us observe that $6L'$ time slots suffice to arrange the departure times of all routes. As far as paths in $\mathcal{P}_c$ are concerned, each one uses one time slot and excludes at most two others, in total using at most $3L_c$ time slots. Similarly, paths in $\mathcal{P}_0$ use at most $3L_0$ time slots.

The distance between time slots in Algorithm 3 is $\frac{T}{6L'}$. We will show that the time distance between any two overlapping routes at any point is not smaller than the time distance between the corresponding time slots. Since the algorithm assigns different time slots to overlapping routes, the minimum time distance is at least $\frac{T}{6L'}$.

Now, observe that no two paths in $\mathcal{P}_c$ can have a time difference smaller than $\frac{T}{6L'}$ in a scheduling produced by the algorithm, because their arrangement is essentially the same as in the case of a chain network.

Suppose we have two overlapping paths $r \in \mathcal{P}_c$ and $r' \in \mathcal{P}_0$. We need to show that their time difference is not less than $\frac{T}{6L'}$ at nodes $s(r)$ and $s(r')$ if these nodes are shared between the two paths, since all trains travel at the same speed.

Let $t_0 = \mathit{timeslot}(r')$ be the time when $r'$ passes through node 0. $r'$ reaches $s(r)$ (if $s(r)$ is contained in $r'$) at time $(\tau(0, s(r))+t_0) \bmod T$, and since $\mathit{stime}(r) = (\tau(0, s(r)) + \mathit{timeslot}(r)) \bmod T$ if $r$ and $r'$ had a time distance of less than $\frac{T}{6L'}$ then they would have been assigned the same time slot, which is a contradiction.

Route $r$ arrives at $s(r')$ at time $\mathit{stime}(r) + \tau(s(r), s(r')) = \mathit{timeslot}(r) + \tau(0, s(r'))$ while $r'$ departs from $s(r')$ at time $\mathit{stime}(r') = \mathit{timeslot}(r') - \tau(s(r'), 0) \equiv_T \mathit{timeslot}(r') - D + \tau(0, s(r'))$. If $r$ and $r'$ had time distance of less than $\frac{T}{6L'}$, then $\mathit{timeslot}(r')$ would have a distance of less than $\frac{T}{6L'}$ from $\mathit{timeslot}(r) + D$ which is also a contradiction.

Finally, let us consider two paths $r$, $r'$ in $\mathcal{P}_0$ to which the algorithm has assigned time slots $\mathit{timeslot}(r)$ and $\mathit{timeslot}(r')$ respectively. Suppose, without loss of generality, that $s(r') > s(r)$. It is clear that since they are assigned different time slots these two paths cannot have a time distance of less than $\frac{T}{6L'}$ at node 0 and therefore neither at node $s(r')$. We now need to show that their time distance is not less than $\frac{T}{6L'}$ at node $s(r)$. We should examine two cases depending on whether $r \prec r'$ or not.

Suppose $e(r') > s(r)$. In that case $r' \prec r$ and $r'$ will be assigned a time slot before $r$. Route $r'$ will reach $s(r)$ at time $\mathit{stime}(r') + \tau(s(r'), 0) + \tau(0, s(r)) \equiv_T \mathit{timeslot}(r') + \tau(0, s(r))$. Route $r$ departs from $s(r)$ at time $\mathit{stime}(r) = \mathit{timeslot}(r) - \tau(s(r), 0) \equiv_T \mathit{timeslot}(r) - D + \tau(0, s(r))$. However, the time distance between $\mathit{timeslot}(r') + D$ and $\mathit{timeslot}(r)$ is at least $\frac{T}{6L'}$, because $r'$ was assigned a time slot before $r$ and $\mathit{timeslot}(r') + D$ was excluded from the list of available time slots.

Let us now assume that $e(r') < s(r)$. In that case the the two paths have only one common segment that starts at $s(r')$ and contains 0. Therefore, the fact that they have been assigned different time slots suffices to guarantee that their time difference is at least $\frac{T}{6L'}$.

□

## 5 PMS in tree networks

In the case of tree networks one might attempt to use Algorithm 2 for spiders, after picking an arbitrary node 0. However, this idea may lead to the production of an infeasible solution. Figure 3 illustrates this situation.

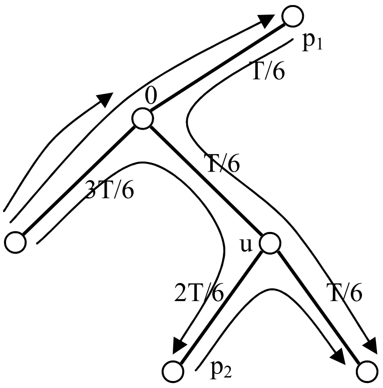

**Fig. 3.** An example showing that Algorithm 2 for spiders does not work for trees. Assuming that path $p_1$ is assigned time slot 0 and path $p_2$ is assigned time slot $\frac{T}{3}$, path $p_1$ collides with path $p_2$ at node $u$ at time $\frac{T}{6}$.

However, if we consider tree networks in which the time needed to travel along each edge is a multiple of $\frac{T}{2}$ it turns out that we can use Algorithm 4. In these networks the following useful property holds.

*Remark 2.* For any three nodes $a, b, c$: $\tau(a,b) + \tau(b,c) \equiv_T \tau(a,c)$.

**Theorem 7.** *An instance of* PMS *in a tree where the time needed to travel along each edge is a multiple of* $\frac{T}{2}$ *admits a solution of value at least* $\frac{T}{k}$ *if and only if the corresponding* PC *instance can be colored with* $k$ *colors.*

*Proof.* For the "if" direction, the following algorithm yields a solution with the desired value.

**Algorithm 4** An algorithm for PMS in tree networks where each edge is a multiple of $\frac{T}{2}$

```
Assume that a coloring with $k$ colors is given.
Pick a root $r_0$ arbitrarily.
Set $t = \frac{T}{k}$ and the available time slots as follows: $0, t, 2t, \ldots, (k-1)t$.
for each path $p$ do
    Let $timeslot(p)$ be the time slot corresponding to the color of $p$.
    Set the starting time of $p$ as follows:
        $stime(p) = (\tau(r_0, s(p)) + timeslot(p)) \bmod T$
end for
```

Assume there are two paths $p$ and $p'$ overlapping on a single edge $e = (u, v)$. Path $p$ reaches node $u$ at time $(\mathit{timeslot}(p) + \tau(r, s(p)) + \tau(s(p), u)) \bmod T = (\mathit{timeslot}(p) + \tau(r, u)) \bmod T$. By the same reasoning path $p'$ reaches node $u$ at time $(\mathit{timeslot}(p') + \tau(r, u)) \bmod T$. Hence, the time distance of the two paths is equal to $(\mathit{timeslot}(p') - \mathit{timeslot}(p)) \bmod T$ which is clearly at least $\frac{T}{k}$.

For the "only if" direction, we pick an arbitrary node $r_0$ and, for each path $p$, we consider the value $\mathit{timeslot}(p) = \mathit{stime}(p) - \tau(r_0, s(p))$. Following the proof of Theorem 3 and using remark 2, we obtain a valid coloring with $k$ colors for the original PC instance.

□

**Corollary 3.** PMS *in trees is NP-hard.*

*Proof.* We will reduce the decision version of PC in trees to the decision version of PMS in trees. Given a PC instance and an integer $k$ we will construct a PMS instance with time distances between nodes equal to one time unit and period $T = 2$. Theorem 7 implies that it is possible to achieve a solution of value at least $\frac{T}{k}$ if and only if the original PC instance can be colored with at most $k$ colors.

□

**Theorem 8.** *Given a $\rho$-approximation algorithm for* PC *in bidirectional trees, Algorithm 4 achieves an approximation ratio of $\frac{1}{\rho}\frac{L}{L+1}$.*

*Proof.* The key observation is that if $\mathrm{OPT}_{\mathrm{PMS}} \geq \frac{T}{\mathrm{OPT}_{\mathrm{PC}-1}}$ then by using algorithm 4 we could achieve a coloring with $\mathrm{OPT}_{\mathrm{PC}} - 1$ colors, which is a contradiction. Therefore $\mathrm{OPT}_{\mathrm{PMS}} < \frac{T}{\mathrm{OPT}_{\mathrm{PC}-1}}$ and the rest of the proof follows along the lines of the proof of Theorem 4.

□

**Corollary 4.** *There is a $\left(\frac{3}{5}\frac{L}{L+1}\right)$-approximation algorithm for* PMS *in trees where the time distances between nodes are multiples of $\frac{T}{2}$.*

*Proof.* By using Theorem 8 and the $\frac{5}{3}$-approximation algorithm of Erlebach et al. [9].

□

## 6 Conclusions

We have introduced the Periodic Metro Scheduling problem, which aims at generating a periodic timetable for a given set of routes and a given time period, in such a way that the minimum time distance between successive trains is maximized.

We have presented exact algorithms for chain and spider networks, and constant ratio approximation algorithms for ring networks, as well as for a special class of tree networks. Some of our algorithms make use of a reduction to Path Coloring. We have left open the question of the approximability of PMS in general tree networks. Another interesting open question is the variation where only the end stations of a route are given and one should determine both a path for each route and a departure time; such a variation applies in topologies that contain cycles, such as rings, grids and trees of rings.

## References

1. L. Anderegg, S. Eidenbenz, M. Gantenbein, C. Stamm, D.S. Taylor, B. Weber, and P. Widmayer. Train Routing Algorithms: Concepts, Design Choices, and Practical Considerations. Proceedings of The Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03), pp. 106–118, 2003.
2. A. Bertossi, P. Carraresi, and G. Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 17:271–281, 1987.
3. M.R. Bussieck, T. Winter, and U. Zimmermann. Discrete optimization in public rail transport. *Math. Program.* 79: 415-444 (1997).
4. M. C. Carlisle and E. L. Lloyd, "On the k-coloring of intervals," *Discrete Applied Mathematics*, vol. 59, pp. 225–235, 1995.
5. G. Carpaneto, M. Dell'Amico, M. Fischetti, and P. Toth. A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19:531–548, 1989.
6. G. Dantzig and D. Fulkerson. Minimizing the number of tankers to meet a fixed schedule. *Nav. Res. Logistics Q.*, 1:217–222, 1954.
7. S. Eidenbenz, A. Pagourtzis, and P. Widmayer. Flexible Train Rostering. Proc. ISAAC 2003, LNCS 2906, pp. 615–624, Springer-Verlag 2003.
8. T. Erlebach, M. Gantenbein, D. Hürlimann, G. Neyer, A. Pagourtzis, P. Penna, K. Schlude, K. Steinhöfel, D.S. Taylor, and P. Widmayer. On the Complexity of Train Assignment Problems. Proc. ISAAC 2001, LNCS 2223, pp. 390–402, Springer-Verlag 2001.
9. T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, and P. Persiano. Optimal wavelength routing in directed fiber trees. *Theoretical Computer Science*, 221:119–137, 1999.
10. P.J. Fioole, L.G. Kroon, G. Maróti, A. Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. CWI Report PNA-E0420, ISSN 1386-3711, 2004. To appear in *European Journal of Operational Research*.
11. Z. Genç. Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitabständen. PhD thesis, Universität zu Köln, Mathematisch-Naturwissenschaftliche Fakultät, Institut für Informatik, 2003. Available at http://kups.ub.uni-koeln.de/volltexte/2003/950/
12. M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In Algorithm Theory - Proceedings SWAT 2004, pages 199–211. Springer-Verlag LNCS 3111, 2004.
13. M. Garey, D. Johnson, G. Miller, and C. Papadimitriou. The complexity of coloring circular arc graphs and chords. *SIAM Journal of Discrete Math*, vol. 1, no. 2, pp. 216-227, 1980.

14. M. Gatto, R. Jacob, L. Peeters, A. Schöbel. The Computational Complexity of Delay Management. In Proc. WG 2005, LNCS 3787, pp. 227–238, Springer 2005.
15. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph, SIAM J. on Computing, 1(1972)180-187.
16. I.A. Karapetian. On the coloring of circular arc graphs. Docladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic, vol. 70(5), pp. 306–311, 1980, in Russian. English translation by D. Gamarnik, D. Williamson and N. Edwards in http://www.orie.cornell.edu/˜nedwards/wdm-routing/
17. V. Kumar. Approximating circular arc coloring and bandwidth allocation in alloptical ring networks. Lecture Notes in Computer Science 1444 (1998), 147-158, Springer-Verlag 1998.
18. L.G. Kroon and L.W.P. Peeters. A Variable Trip Time Model for Cyclic Railway Timetabling. *Transportation Science* vol. 37 no. 2 pp. 198-212, INFORMS, 2003.
19. C. Liebchen and R.H. Möhring. A Case Study in Periodic Timetabling. *Electr. Notes Theor. Comput. Sci.* 66(6), 2002.
20. C. Liebchen. A Cut-Based Heuristic to Produce Almost Feasible Periodic Railway Timetables. In Proc. WEA 2005, LNCS 3503, pp. 354-366, Springer-Verlag 2005.
21. A. Löbel. Optimal vehicle scheduling in public transit *PhD thesis*, *TU Berlin*, 1998.
22. Milena Mihail, Christos Kaklamanis, Satish Rao: Efficient Access to Optical Bandwidth - Wavelength Routing on Directed Fiber Trees, Rings, and Trees of Rings. FOCS 1995: 548-557.
23. P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4), 550–581, (1989).
24. A. Schöbel. A model for the delay management problem based on mixedinteger-programming. In Christos Zaroliagis, editor, *Electronic Notes in Theoretical Computer Science*, volume 50. Elsevier, 2001.
25. A. Schrijver. Routing and Timetabling by Topological Search. *Documenta Mathematica*, Extra Volume ICM III (1998), 687-695.
26. A. Schrijver. Minimum circulation in railway stock. *CWI Quarterly*, 6(3):205–217, 1993.