Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 168 (2006) 403–424
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

On solving multi-type railway line planning problems

Jan-Willem Goossens [a,*], Stan van Hoesel [a], Leo Kroon [b]

[a] Department of Quantitative Economics, University of Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands
[b] Rotterdam School of Management, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Received 14 May 2003; accepted 19 April 2004
Available online 2 July 2004

Abstract

An important strategic element in the planning process of a railway operator is the development of a line plan, i.e., a set of routes (paths) on the network of tracks, operated at a given hourly frequency. The models described in the literature have thus far considered only lines that halt at all stations along their route. In this paper we introduce several models for solving line planning problems in which lines can have different halting patterns. Correctness and equivalence proofs for these models are given, as well as an evaluation using several real-life instances.

Keywords: Integer programming; Combinatorial optimisation; Railway transportation

1. Introduction

The planning problem faced by every railway operator consists of several consecutive stages, ranging from strategic decisions concerning, e.g., the acquisition of rolling stock, to operational traffic control. Strategic problems are largely driven by estimates for the long-term demand. Together with infrastructure data, such as the railway tracks and stations, these demand data are input for the strategic line planning problem considered in this paper. It involves the selection of paths in the railway network on which train connections are operated. Thus, the line planning problem focuses on determining a subset of all possible paths (lines) that together make up the line plan, such that the provided transportation capacity is sufficient to meet the passenger demand. Relevant objectives are the provided service towards the passengers and the operational costs for the railway operator.

* Corresponding author. Tel.: +31 0 6 10326976; fax: +31 0 43 3884874.
E-mail addresses: j.goossens@t75.nl (J.-W. Goossens), s.vanhoesel@ke.unimaas.nl (S. van Hoesel), l.kroon@fbk.eur.nl (L. Kroon).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.04.036

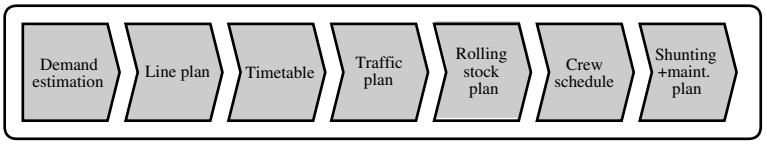

Fig. 1. The different stages in the planning process.

Successive decision stages, as shown in Fig. 1, are the more detailed planning problems such as the construction of timetables [9,10,14], traffic planning (route assignment, platform assignment) [16], rolling stock planning [1,12,13], crew scheduling [3,8] and shunting planning [5,6,15].

Besides the operated paths, a line plan also specifies the hourly frequencies of the lines and their halting patterns. The halting pattern defines the stations along a line's route at which it halts. Halting patterns for train lines can be divided into classes, called *types*, such as Regional, Interregional and Intercity. The line planning models described in the literature have thus far considered all lines to be of the same type (see [2,4,7]). Solving problems with more than one type was done by a priori assigning the passengers to the different train types, thus splitting the original problem into separate problems for every type. This allocation was determined, for example, by the procedure System Split (see [11]). In this paper, we introduce several generalisations of the previous models to simultaneously solve cost-optimising line planning problems with multiple train types.

Traditionally, the objective when constructing a line plan has been to find a set of lines that maximises the number of direct travellers, as in Bussieck [2]. This is an obvious objective from a service perspective, since it maximises the number of travellers that do not have to change trains during their journey. However, this objective tends to generate geographically long train lines, since the longer the lines, the more direct connections are provided. Such long lines often have large fluctuations in the number of passengers on different parts of their route. Therefore, long train lines can result in unused capacity on the less busy tracks, and can thus be inefficient and expensive. As an alternative objective, similar to Claessens et al. [4], Bussieck [2] and Goossens et al. [7], this paper focuses on models for minimising the operational costs of a line plan.

The next section recalls the single-type line planning problem, and introduces new definitions for the multi-type model. In Section 3 we discuss how to formulate the multi-type model by using an intermediate problem, called the edge capacity problem. For this problem we consider a number of model formulations in Section 4. Apart from a multi-commodity flow formulation, we develop two alternative mathematical formulations and prove their equivalence. In Section 5 we describe a computational study, based on instances of the Dutch railway operator NS Reizigers.

2. Modelling

The concept of a line is fundamental in a railway system for passenger transportation. A line specifies a route between an origin and a destination station and the subsequent stops, combined with an operated hourly frequency. The line plan is the set of operated lines. The line plan does not incorporate the exact timetable for the operated lines, though we assume that the timetable will be cyclic with a cycle time of one hour, i.e., that the line plan is repeated every hour. Note that this still allows for lines to be operated with a frequency of for example 2, i.e., twice per hour. The models described here focus on finding a line plan that minimises the induced operational costs (see [2,4,7]).

Before discussing models for simultaneously solving multi-type line planning problems (MLPP), let us first recall the single-type cost-optimising line planning problem (LPP).

2.1. The cost-optimising line planning problem

To formulate LPP, we introduce the track graph $G=(V,E)$ consisting of the set of vertices (stations) V, and the connecting edges (tracks) E. To later distinguish between different kinds of edges, we refer to the edges in the track graph as *track edges*. In addition, we are given a set of potential lines L. Every line $l\in L$ corresponds to a set of track edges that make up a simple path between the end vertices of l. We later define the notion of *types* of lines, but for now we consider all lines in L to be of the same type. The consecutive stops of line l are given by the stations corresponding to the vertices along its path. For every line we have to decide whether to deploy it, and, if so, at what hourly frequency. The operational hourly costs of a line plan can be described by a function of fixed and variable costs per used train and per carriage. To estimate these costs, we have to decide for every line not only at what hourly frequency it will be operated, but also with how many carriages per train. Indeed, the rolling stock is one of the largest cost drivers of a railway operator.

The set of possible frequencies for the lines is denoted by $F\subset\{1,2,\ldots\}$, the possible number of carriages per train by $C\subset\{1,2,\ldots\}$. For formulating the line planning problem as an integer linear programming problem, we introduce a binary variable for every $(l,f,c)\in N$, with the set of triples as $N:=L\times F\times C$. Every $i\in N$ is used for referring to a particular combination (l_i,f_i,c_i). For convenience, we also introduce the set $N(e)\subseteq N$ for every track edge e as $N(e)=\{i\in N:e\in l_i\}$. Now the LPP can be formulated as follows:

$$z_{\mathrm{LPP}} = \min \quad \sum_{i\in N} w_i x_i \tag{1}$$

$$\text{s.t.} \quad \sum_{i\in N(e)} f_i c_i x_i \geqslant h^e \qquad \forall e\in E, \tag{2}$$

$$\sum_{i\in N\mid l_i=l} x_i \leqslant 1 \qquad \forall l\in L, \tag{3}$$

$$x_i \in \{0,1\} \qquad \forall i\in N, \tag{4}$$

where the decision variables x_i for $i=(l_i, f_i, c_i)\in N$ are used to model

$$x_i = \begin{cases} 1 & \text{if line } l_i \text{ is operated at frequency } f_i, \text{ with } c_i \text{ carriages per train,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The objective function coefficients w_i represent, in general, the costs of operating the line associated with variable x_i. In this paper, the induced operational costs are defined only on the costs per carriage. The capacities of lines in a line plan are assumed to be identical in both directions, i.e., from origin to destination, and vice versa. This assumption is widely adopted by many authors (see [2,4]). To operate line l at an hourly frequency of f it is necessary to run at least $\lceil cp_l\cdot f\rceil$ trains, where cp_l is the total time in minutes that a train of line l needs to complete its journey, return to its origin station and start its next trip, divided by sixty. Now, the hourly costs of operating a line l of type t at frequency f, and with c carriages per train can be calculated as

$$w_i = \lceil cp_l\cdot f\rceil\cdot c\cdot w_{\mathrm{fix}}^{\mathrm{car}}(t), \tag{6}$$

where $w_{\mathrm{fix}}^{\mathrm{car}}(t)$ represents the hourly costs of operating one carriage of a line of type t. Note that we consider all carriages for lines of one specific type to be identical.

Restrictions (2) impose the lower bound h^e on the number of carriages crossing edge e per hour. Typically, h^e is the smallest number of carriages necessary for transporting all passengers that want to cross edge e. The constraints (3) ensure that every line is operated in at most one configuration. Note that, in contrast to other authors, we do not consider any lower bound restrictions on the number of train connections between adjacent stations.

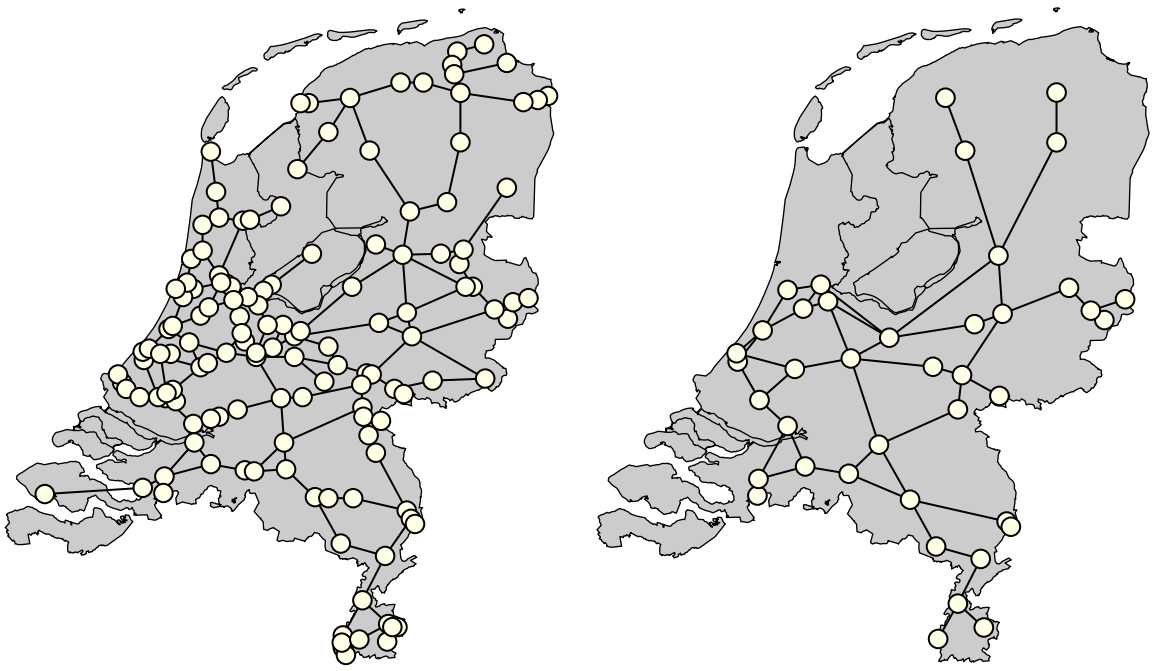

Fig. 2. Two single-type LPP instances: the regional train instance SP97AR (left), and the intercity train instance SP97IC.

The track graphs for two single-type instances are shown in Fig. 2. As mentioned in the introduction, to solve the line planning problems for the individual train types, such as, e.g., intercity trains, the overall passenger flows are a priori split into demands for the separate systems.

The essential difference between LPP and the multi-type problem MLPP is the integration of the different networks, and thus the possibility of the model to distribute the passengers over the trains of the different types. An example of a new kind of restriction in a multi-type setting, where trains no longer halt at every station, is to ensure that passengers are not assigned to trains that pass their destination station without stopping there. To arrive at a formulation for MLPP, we are going to present a simplified version of this problem, called the edge capacity problem. Where MLPP decides on the provided capacities for the lines, in the edge capacity problem the decision variables model the supplied capacities for individual edges. First, we introduce several new concepts and notations.

2.2. Definitions and notation

On the previously given track graph $G=(V,E)$, we define a *commodity* $k=(s^k,t^k)\in V\times V$, that can be seen as travellers that want to travel from their source station s^k to their destination t^k. A commodity is not allowed to use just any arbitrary path through the network. Instead, every commodity k is restricted to use the track edges of a given simple path $P_k\subseteq E$ between s^k and t^k. This is comparable to the restriction enforced by the ticket regulations. In general this route is the shortest path. The assumption that there is exactly one fixed route is not important. The essence is that the route is known for every traveller. The demand for commodity k, i.e., the number of travellers that want to travel from s^k to t^k, is given by its entry $H^{s^kt^k}$ in the square demand matrix H. This demand for commodity k is also denoted by H^k. The matrix H, also called the Origin–Destination (OD) matrix, is assumed to be symmetric, i.e., that it has the property

that $H^{st}=H^{ts}$ for all s and t. The developed models can, however, easily be adapted to suit instances for which this assumption does not hold.

Every vertex in the track graph is of a certain *type*. If we denote the set of available types $T=\{1,\ldots,T_{\max}\}$, then every vertex $v\in V$ is of type $t_v\in T$. In most instances, these types represent the sizes of the stations: $t_v=1$ for stations in villages up to $t_v=T_{\max}$ for stations serving large metropolitan areas. Most real-life instances consider three types of stations and train lines. These are usually referred to as Regional (R) or stop trains for type 1, Interregional (IR) for type 2, and Intercity (IC) for type 3. A similar categorisation is also made for the train lines that will be operated on the network. The route of a train line through this network is a path-shaped collection of connected tracks. The type of a train line determines the stations along the line's route at which the line halts. Train lines of type 1, for example, halt at all stations they pass. Lines of type 2 skip the small stations of type 1 etc. In general, a train line of type t halts at all stations v along its route with a type $t_v \geqslant t$.

Example 2.1. Consider the track graph in Fig. 3. The network is described by the connected graph $G=(V,E)$, where $V:=\{a,b,c,d,v,w\}$, and $E:=\{\{a,b\},\{b,c\},\ \{c,d\},\ \{d,v\},\{v,w\}\}$. The type of a station is given by the number below its vertex. Thus, $t_a=1$, $t_b=3$ etc. In this graph G, we have defined three lines of different types. Line 1 of type 1, going from station a to station w, halting at all stations in between. Line 2 of type 2, from station b to station w, halting only at stations c and v. Line 3 of type 3 that does not halt at any station, apart from its origin station b and destination station v. The halting patterns are also shown by the vertical dashes in the lines that represent the routes of the train lines.

Notice in Example 2.1 that travellers using train line 3 of type 3 to travel from b to v will not halt at any of the stations in between. We could thus introduce an edge $\{b,v\}$ of type 3 to show that, due to the types of the stations in between b and v, train lines of type 3 will not stop at any of the stations between b and v. That is, they will use edge $\{b,v\}$ instead. In general, we construct from the track graph G its *type* graph $G^T=(V,E^T)$. With an identical set of vertices, the difference between G and G^T lies in the set of edges. In the type graph we introduce $T_{\max}$ sets of edges. See Example 2.2. We refer to edges of the type graph as *type edges*.

Example 2.2. The track graph G given in Fig. 3 can be transformed into the type graph $G^T=(V,E^T)$ displayed in Fig. 4. Note that the structure of the type graph depends on the types of the stations, not of the lines.

The mapping of the edges in the type graph G^T to the original track edges in G is done through the definition of the *route* of a type edge e. The route $R(e)\subseteq E$ is the simple path in the original track graph that is covered by the type edge e. So, in the example above, with type 3 edge $\{b,v\}$, we have that $R(\{b,v\})=\{\{b,c\},\ \{c,d\},\ \{d,v\}\}$. The overall set of type edges E^T of the type graph is the union of all the sets of edges of a type t, so $E_T:=\bigcup_{t\in T}E_t^T$. The edge set E_t^T contains all type edges of type t. We assume that the lowest set of type edges is equal to the set of track edges, i.e., $E_1^T=E$. Every edge e of type t, i.e.,

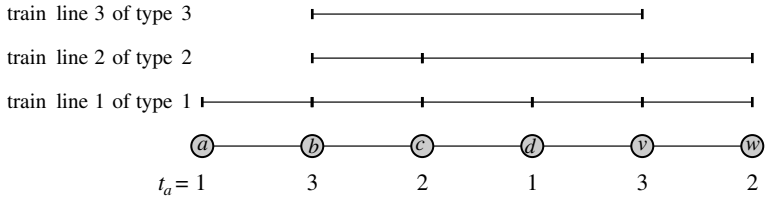

Fig. 3. The track graph G showing the types of the stations and several train lines.

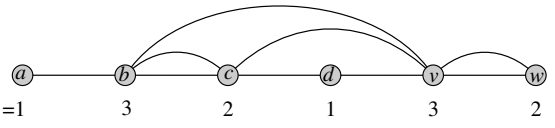

Fig. 4. The type graph, based on the track graph from Fig. 3.

every $e \in E_t^T$ satisfies that its route $R(e)$ contains only internal vertices i for which $t_i < t$. We say that a type edge g is *covered* by a type edge f if $R(g) \subseteq R(f)$. Again, e.g., in Example 2.2 the edge $\{c, v\}$ is covered by $\{b, v\}$.

The sets E_t^T for all $t \in T$, together with the corresponding routes of the type edges, are part of the problem input. In most cases the route for any pair of nodes v and w describes the shortest path from v to w. The sets E_t^T need not be exhaustive, i.e., not every pair of vertices $v, w \in V$ for which there exists a simple path has to be present in E_t^T. The graph G^T can be a multi-graph, in the sense that some type edge $\{v, w\} \in E_t^T$ and $\{v, w\} \in E_{t'}^T$ for $t \neq t'$. This is shown in Example 2.2 by the type edges $\{b, c\}$ and $\{v, w\}$.

We assume that the route definitions are consistent, i.e., if the route $R(e)$ of type edge $e = \{i,j\} \in E^T$ contains two vertices v and w for which there exists a type edge $f = \{v, w\} \in E^T$, then also $R(f) \subseteq R(e)$. This is illustrated in Fig. 4 by the edge $e = \{b, v\}$ of type 3, and the type 2 edge $f = \{c, v\}$. In addition, we assume that if there exists an edge g of type $t > 1$ whose route contains a track edge e, then there is also a type edge f of type $t - 1$ whose route is contained in that of g, and that also covers e: $e \subseteq R(f) \subseteq R(g)$. It would be the same to assume that the complete route $R(g)$ of g can be covered by the routes of edges of type $t - 1$, that are all contained in $R(g)$. Thus, for the type edge $g = \{b, v\}$ we assume the presence of the type 2 edges $\{b, c\}$ and $\{c, v\}$. Both of these assumptions are not very restrictive.

Using the edges of the type graph, we introduce the set $P_k^T \subseteq E^T$ of type edges for every commodity k. These paths consist of the type edges that make up the best (highest type) possible route across a commodity's path P_k from s^k to t^k. Formally, for all type edges $e \in E_t^T$ it should hold that

$$e \in P_k^T \iff R(e) \subseteq P_k \text{ and } \nexists t' > t : \exists f \in E_{t'}^T : R(e) \subseteq R(f) \subseteq P_k. \tag{7}$$

Hence, in the graph in Fig. 4 the best-edge path for $k = (a, w)$ is $P_k^T = \{\{a, b\}, \{b, v\}, \{v, w\}\}$ where $\{a, b\} \in E_1^T, \{b, v\} \in E_3^T$ and $\{v, w\} \in E_2^T$.

3. Formulating the multi-type line planning problem

This section describes the first step in extending the formulation for LPP to be able to model multiple train types simultaneously in the MLPP.

Once again, every line l corresponds to a route (a simple path) through the track graph G. The halting pattern of a line l, i.e., the stations along the route of l at which it stops, is dictated by its type $t_l \in T$, similar to the edges in the type graph. Thus, l is said to *use* a simple path of type edges in the type graph G^T, namely the path of all type edges $e \in E_t^T$ for which $R(e) \subseteq l$. Again, as in LPP, for every line we have to decide whether to deploy it and, if so, at what hourly frequency, and with how many carriages per train. Now, however, the possible frequencies and number of carriages of a line depend on its type: valid frequencies and capacities of lines of type t are given by $F(t) \subset \mathbb{N}$, and $C(t) \subset \mathbb{N}$ respectively. The set N of triples is now defined as $N := \{(l, f, c) | l \in L, f \in F(t_l), c \in C(t_l)\}$.

As in (5), a binary variable x_i for $i \in N$ is used to indicate whether line l_i is operated at frequency f_i, with c_i carriages per train. On every type edge e in the network, the total capacity provided by all lines that use this edge is given by

$$\sum_{i \in N | l_i \text{ uses } e} \lambda(t_{l_i}) f_i c_i x_i, \tag{8}$$

where $\lambda(t)$ represents the capacity, in number of passengers, of one carriage of lines of type t. If a line is selected to be in the line plan at a certain configuration, then it thus provides capacity along all the edges in the type graph G^T that it uses.

Instead of formulating MLPP directly, we consider a simplified problem called the edge capacity problem (ECP). The ECP is described on the track graph G and the associated type graph G^T. The problem is to assign enough capacity to individual edges in the type graph G^T, such that all commodities can be transported simultaneously, while minimising some objective function of the allocated capacity. As such, MLPP is a generalisation of ECP. We are going to present several formulations of ECP in which variables $x(e) \in \mathscr{C}$ are used to represent the amount of capacity that is assigned to type edge e. It is clear that if we have formulated ECP, then MLPP can be formulated by substituting (8) of MLPP for every variable $x(e)$ of ECP. In addition, the domain restrictions $x(e) \in \mathscr{C}$ have to be replaced by

$$\sum_{i \in N | l_i = l} x_i \leqslant 1 \qquad \forall l \in L,$$
$$x_i \in \{0,1\} \qquad \forall i \in N,$$

as in (3) and (4). The objective function $\sum_{e \in E^T} f(x(e))$ of ECP can be replaced by $\sum_{i \in N} w_i x_i$ for MLPP, where the weights w_i are defined as in (6).

4. Formulations for the edge capacity problem

We now present three different formulations for ECP.

4.1. The multi-commodity flow formulation (MCF)

Let us introduce two directed graphs, similar to the track graph and the type graph. First, $D=(V,A)$ is constructed from the track graph G using the arc set A which contains a forward arc (i,j) and a backward arc (j,i) for every track edge $\{i,j\} \in E$. Second, the directed graph $D^T=(V,A^T)$ is built similarly from the undirected type graph G^T by replacing every type edge in E^T by two opposing arcs in A^T. For dealing with these directed graphs we define $\vec{R}(a) \subseteq A$ as the directed simple path for an arc $a=(i,j) \in A_t^T$ similar to $R(e)$ for the corresponding type edge $e=\{i,j\} \in E_t^T$. The prescribed path $P_k \in E$ for commodity k in the original graph is represented by the directed simple path $\vec{P}_k \subseteq A$.

In general, a feasible multi-commodity flow satisfies the flow conservation constraints

$$\sum_{j:(i,j) \in A^T} F_{ij}^k - \sum_{j:(j,i) \in A^T} F_{ji}^k = b_i^k \qquad \forall i \in V,\ \forall k \in V \times V, \tag{9}$$

where the flow variables F_{ij}^k represent the number of passengers of the commodity k that use arc $(i,j) \in A^T$ through the directed type graph D^T. The right-hand sides b_i^k are chosen such that

$$b_i^k = \begin{cases} H^{s^k t^k} & \text{if } i = s^k, \\ 0 & \text{if } s^k \neq i \neq t^k, \\ -H^{s^k t^k} & \text{if } i = t^k. \end{cases}$$

The MCF can thus be modelled as follows:

$$\min \quad \sum_{e\in E^T} f(x(e)) \tag{10}$$

$$\text{s.t.} \quad x(e) \geqslant \sum_{k} F_{ij}^{k} \qquad \forall t \in T,\ \forall (i,j) \in A_t^T,\ e=\{i,j\} \in E_t^T, \tag{11}$$

$$\sum_{j|(i,j)\in A^T} F_{ij}^{k} - \sum_{j|(j,i)\in A^T} F_{ji}^{k} = b_i^k \qquad \forall i \in V,\ \forall k \in V\times V, \tag{12}$$

$$F_{ij}^{k} = 0 \qquad \forall k \in V\times V,\ \forall (i,j) \in A^T : \vec{R}((i,j)) \not\subseteq \vec{P}_{s^k t^k}, \tag{13}$$

$$F_{ij}^{k} \in \mathbb{N} \qquad \forall k \in V\times V,\ \forall (i,j) \in A^T, \tag{14}$$

$$x(e) \in \mathscr{C} \qquad \forall e \in E^T. \tag{15}$$

From the construction of the directed type graph D^T it is evident that there is an exact 1-to-2 relation between an edge $e=\{i,j\}\in E_t^T$ for some type t, and a pair of arcs (i,j) and (j,i), both in A_t^T (and vice-versa). This relation is used in constraints (11) to enforce that the capacity assigned to type edge e, $x(e)$, is at least as large as the flow across both related arcs. The combined capacity of all lines connecting two stations should be at least as large as the flow in either direction. Consider, for example, a network with two stations v and w. If 50 people want to travel from v to w, and 60 from w to v, then the combined capacity of the lines that connect v and w should be at least $\max\{50,60\}=60$.

The restrictions (12) are the flow conservation constraints for every vertex. Restrictions (13) enforce that travellers between a and b have to travel using arcs that are within their predetermined path $\vec{P}_{ab}$. In the directed type graph D^T, we thus restrict k to use only arcs (i,j) for which $\vec{R}((i,j))\subseteq\vec{P}_k$. Finally, the set of feasible values for $x(e)$ is given by the set $\mathscr{C}\subset\mathbb{N}$, which represents the possible capacities of edges.

We will now describe two lemmas that will be used to preprocess problem instances, and to prove the equivalence of alternative models. Let us first show that a commodity $k=(n,m)$ can be split into a number of partial commodities if its path $\vec{P}_k^T$ consists of more than one arc. Every feasible flow for these partial commodities can be recombined to a feasible flow for the original commodity k, while the reverse also holds.

Example 4.1. Let us preview the commodity decomposition principle on the track graph G and the type graph G^T used in Example 2.2. Fig. 5 first of all shows the directed graph D^T based on G^T. In addition, it also shows how the commodity $k=(a,w)$ and its best path $\vec{P}_{aw}^T$ are decomposed from $\vec{P}_{aw}^T=\{(a,b),(b,v),(v,w)\}$ to three separate commodities $k_{(a,b)}$, $k_{(b,v)}$ and $k_{(v,w)}$ and the three best paths $\vec{P}_{k(a,b)}^T=\{(a,b)\}$, $\vec{P}_{k(b,v)}^T=\{(b,v)\}$ and $\vec{P}_{k(v,w)}^T=\{(v,w)\}$. The commodity decomposition Lemma 4.2 shows that if we have a feasible flow for the three separate commodities, then it is possible to recombine it into a feasible flow for the original commodity k, and vice versa.

Lemma 4.2 (Commodity Decomposition). *Given a commodity k with demand H^k and arc set $\vec{P}_k^T$, consider the following decomposition. Every feasible flow for a commodity k with demand H^k can be split into a feasible flow for $|\vec{P}_k^T|$ new commodities k_f, with $f\in\vec{P}_k^T$, for which the demand is $H^{k_f}=H^k$ and with best-arc set $\vec{P}_{kf}^T=\{f\}$. The reverse—combining of the flows—results again in a feasible flow for k.*

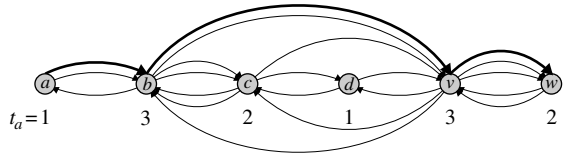

Fig. 5. The digraph D^T based on the network in Fig. 4.

Proof. For all arcs $g=(i,j)\in A^T$ that can be used by k, i.e., for which $\vec{R}(g)\subseteq\vec{P}_k$, there exists an arc $f=(n,m)\in\vec{P}_k^T$ in which g is contained ($\vec{R}(g)\subseteq\vec{R}(f)$). We will prove this lemma by showing that setting the flows equal to

$$F_{ij}^{k_f}=F_{ij}^{k}\qquad\forall f,\forall(i,j)|\vec{R}((i,j))\subseteq\vec{R}(f)$$

and vice versa, satisfies the flow balance restrictions of both instances. Let us start with proving the decomposition. Note that, constructed in this way, it is sufficient to show that the flow balance constraints for commodity k_f are satisfied at both endpoints of an arc $f=(n,m)\in\vec{P}_{kf}^T$:

$$\sum_{i|(n,i)\in A^T}F_{ni}^{k_f}=H^k=H^{k_f}\quad\text{and}\quad\sum_{j|(j,m)\in A^T}F_{jm}^{k_f}=H^k=H^{k_f}.$$

Equality holds in both cases because f is one of the best-path arcs for k, since this implies that the total flow of k (and of k_f) uses f or arcs covered by f. For internal nodes in $\vec{R}(f)$, the flow balance constraints are already satisfied because F_{ij}^k is a feasible flow. Next, consider the reverse, i.e., that the combined flow is a feasible flow for k. This is true since the flow balance constraints are satisfied because both the total amount of incoming flow via f_1 ($H^{k_{f_1}}$) and outgoing flow via f_2 ($H^{k_{f_2}}$) are equal to H^k by construction. □

The application of the decomposition part of Lemma 4.2 for all commodities, will result in many commodities with the same origin, destination, and prescribed path. The following lemma shows that these similar commodities can be aggregated, thereby reducing the total number of commodities in the system.

Lemma 4.3 (Commodity Aggregation). *Consider two commodities $k_1=(n,m)$ and $k_2=(n,m)$ with identical prescribed paths $\vec{P}_{k_1}=\vec{P}_{k_2}$. The demands for the commodities are given by H^{k_1} and H^{k_2}. Both commodities can be replaced by a new commodity k with demand $H^k=H^{k_1}+H^{k_2}$ and path $\vec{P}_k=\vec{P}_{k_1}=\vec{P}_{k_2}$. Conversely, every feasible flow for k can be disaggregated into feasible flows for k_1 and k_2.*

Proof. Consider a feasible flow for commodity k with demand $H^k=H^{k_1}+H^{k_2}$. Construct two separate flows k_1 and k_2 by labelling H^{k_1} of the leaving flow units in n red, and H^{k_2} of them blue. Clearly, these flows are still feasible flows. The reverse is shown by removing the labels from both commodities. □

By the previous two lemmas, we can assume that all commodities $k=(n,m)$ in an instance of MCF have the property that $\vec{P}_k^T=\{(n,m)\}$. Note that this does *not* imply that an arc (n,m) can only be used by one commodity, since commodities are still allowed to be routed using all arcs in their prescribed path.

The results of both lemmas hold because the ECP is only interested in finding a capacity assignment that minimises the total cost. Any information about the flows according to the original routes of the passengers is lost by decomposing and aggregating the commodities.

Example 4.4. Let us review the MCF problem on the graph displayed in Fig. 4 on page 7. Originally, this problem contained $6\times5=30$ different commodities, i.e., one for every pair of vertices. After applying both of the lemmas above, we are left with at most $|A^T|=2|E^T|=18$ commodities. However, the type 1 edges $\{b,c\}$ and $\{v,w\}$ can never be part of a best path because of the similar type 2 edges. Therefore, the number of commodities can be reduced to 14.

Next, we use the previous two lemmas to show that we can assume that there exists an optimal flow that is symmetric. This is shown by using induction on the number of train types $T_{\max}$. In the induction step, where we assume that we can construct a symmetric solution for $T_{\max}=t^*$, we show how to transform a non-symmetric flow across the arcs of type $T_{\max}=t^*+1$ into a symmetric flow.

Corollary 4.5. *Consider an arbitrary instance of MCF. If, for some arc (i,j) of type $T_{\max}$ there exists a commodity k for which $(i,j)\in\vec{P}_k^T$, then k is also the only commodity with this property.*

Proof. Recall the definition of P_k^T, and thus of $\vec{P}_k^T$, in (7). Since arc (i,j) is of type $T_{\max}$, there do not exist any arcs $(i',j') \neq (i,j)$ of any type $t > T_{\max}$. □

Theorem 4.6 (Symmetric Flow). *If the demand matrix H is symmetric, then, for any solution (X^*,F^*) of MCF, there exists a solution (X^*,F) with the same objective function value, and with the property that $F_{ij}^{nm} = F_{ji}^{mn}$, i.e., that F is a symmetric flow.*

Proof. We prove this theorem using induction on the number of types $T_{\max}$. Initially, consider $T_{\max}=1$. Since there is only one type, and the prescribed path is simple, every commodity has one unique path in the type graph from its origin to its destination. Therefore, in case $T_{\max}=1$, F^* will be symmetric, given that H is symmetric.

Next, assume that the theorem holds for $T_{\max}=t^*$. We show that this implies that it also holds for $T_{\max}=t^*+1$. From Corollary 4.5 we know that for every arc of type $T_{\max}$, there is at most one commodity that is allowed to use this arc. If such a commodity does not exist, then we are done. Hence assume that there exists one commodity for arc (i,j) and one for arc (j,i). Thus, for the type edge $\{i,j\} \in E_{t^*+1}^T$, Eq. (11) tells us

$$x(\{i,j\}) \geqslant \sum_k F_{ij}^k = F_{ij}^{ij} \quad \text{and} \quad x(\{i,j\}) \geqslant \sum_k F_{ji}^k = F_{ji}^{ji}.$$

Suppose the two opposing flows defined for this type edge are not symmetric. So, without loss of generality, assume $F_{ij}^{ij} < F_{ji}^{ji}$. We can now find $F_{ji}^{ji} - F_{ij}^{ij}$ units of flow of commodity (i,j) and, according to Lemma 4.3, reassign them to the arc (i,j), making the flow on (i,j) and (j,i) equal. The capacity restriction for $x(\{i,j\})$ in (11) will still be satisfied. Since we have only redirected flow away from the other arcs that could be used by (i,j), this also holds for the type edges below $\{i,j\}$. The resulting flow is feasible for MCF, and is symmetric on all edges of type t^*+1 by repetition. Now, let us construct a new MCF instance with only t^* types from which the arcs of type t^*+1 have been removed and the demands for commodities (i,j) have been decreased by F_{ij}^{ij} for every $(i,j) \in A_{t^*+1}^T$. Clearly, the previous flow on all but the arcs of type t^*+1 is a feasible flow for this new instance. Therefore, by the induction hypothesis, this MCF can be made symmetric. The overall cost will now be $\sum_{e \in E_{t^*+1}^T} f(x^*(e)) + \sum_{t \leqslant t^*} \sum_{e \in E_t^T} f(x^*(e)) = \sum_{e \in E^T} f(x^*(e))$. □

In view of Theorem 4.6 we will no longer distinguish the commodities (n,m) and (m,n), or the arcs (i,j) and (j,i), since we have shown that we can assume that $F_{ij}^{nm} = F_{ji}^{mn}$. Therefore, we will no longer use the directed graphs D and D^T.

Before we introduce alternative model formulations for ECP, let us first make some general remarks about the structure of the undirected track graph and the type graph.

Lemma 4.7. *Consider a track graph $G=(V,E)$ that is a path. Now, for every track edge $e \in E$ and type $t \in T$, there is at most one type edge f of type t for which $e \in R(f)$.*

Proof. Without loss of generality, we can rename the vertices and track edges of G such that $V=\{1,\ldots,n\}$ and $E=\{\{v,v+1\} | v \in \{1,\ldots,n-1\}\}$, since G is a path. The proof is by contradiction. Assume that, for an arbitrary type t, there are two distinct type edges $f=\{v,w\}$ and $g=\{i,j\}$ in E_t^T that cover e. Without loss of generality, we can assume not only that $v<w$ and $i<j$ and $v \leqslant i<w$, but also that either $w<j$ (crossing) or $j<w$ (non-crossing). Note that if $w=j$, then we could reverse the numbering of the vertices. The first case implies that $t_w \geqslant t$ since $f \in E_t^T$, while the fact that w is an internal vertex of $R(g)$ implies that it is of type less than t. Similar reasoning can be applied in the second case. □

Corollary 4.8. *For any two distinct type edges f and g both of type t with $R(f) \cap R(g) \neq \emptyset$, the graph induced by $R(f) \cup R(g)$ is not a path.*

Proof. Assume that the graph induced by the track edges in $R(f) \cup R(g)$ is a path. Since $R(f) \cap R(g) \neq \emptyset$, we know that there is at least one track edge e for which $e \in R(f)$ and $e \in R(g)$. This is not possible according to Lemma 4.7. □

Corollary 4.9. *There does not exist a type edge h of type t' that covers two type edges f and g of type $t < t'$ for which $R(f) \cap R(g) \neq \emptyset$.*

Proof. By contradiction, assume that $R(h)$ is a path. Clearly, this implies that also $R(f) \cup R(g)$ is a path. However, this contradicts Corollary 4.8. □

Lemma 4.10. *Consider a track edge $e \in E$, and an arbitrary type edge g of type t with $e \in R(g)$. Now, for every type $t' < t$ there exists a unique type edge f of type t' with $e \in R(f) \subseteq R(g)$.*

Proof. First, consider the case where $t' = t - 1$. Now, existence is immediate from the assumptions. Uniqueness follows from Corollary 4.9. Moreover, since the existence and uniqueness also hold for this type edge of type t', there thus exists a unique type edge for every type $t' < t$, by repetition. □

Thus, we have shown that there cannot exist a type edge h that covers two overlapping type edges $f, g \in E_t^T$ of the same type, simply because the original track edges covered by f and g cannot be a path according to Corollary 4.8.

4.2. The integer programming formulation (IP_{XY})

Solving ECP problems using the MCF formulation requires a large number of variables and restrictions. It introduces a flow variable for all the available arcs in the path for every commodity, requiring flow conservation constraints for all the nodes along this path. We will now describe an integer programming model, using fewer variables, and show the equivalence of both models.

Compared to the MCF formulation with its completely disaggregated flow, the IP_{XY} formulation is based on constraining only the capacities of the edges in the type graph. For ease of notation, let us introduce $\tilde{H}(e)$ as the number of travellers for whom type edge e is part of their best path P_k^T. Thus, we define $\tilde{H}(e) = \sum_{k|e \in P_k^T} H^k$ for every type edge e. We also introduce additional variables y_e for every type edge $e \in E_t^T$ with $t > 1$. They represent the number of travellers over all pairs (a, b) that could have used type edge e across this particular part of their path P_{ab}, but do not. Capacity will be reserved for them on the underlying type edges.

Example 4.11. Consider the type graph in Fig. 6. The (b, v)-travellers can either use the type 3 edge f from b to v, or they are assigned to the two underlying type 2 edges $\{b, c\}$ and $e = \{c, v\}$ using the variable y_f. Whether they will actually use these type 2 edges depends on the individual values of the variables $y_{\{b,c\}}$ and y_e through which they can be assigned to the underlying type 1 edges. In this example, the capacity restrictions for the type 3 edge f and for the type 2 edge e will be

$$x(f) \geqslant \tilde{H}(f) - y_f \quad \text{and} \quad x(e) \geqslant \tilde{H}(e) + y_f - y_e.$$

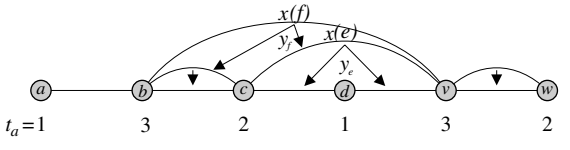

Fig. 6. Travellers are possibly assigned to underlying type edges.

The idea illustrated in Example 4.11 can be generalised to the following model, referred to as IP_{XY}.

$$\min \quad \sum_{e\in E^t} f(x(e)) \tag{16}$$

$$\text{s.t.} \quad x(e) \geqslant \tilde{H}(e) + \sum_{f\in E_2^T | R(e)\subseteq R(f)} y_f \qquad \forall e \in E_1^T, \tag{17}$$

$$x(e) \geqslant \tilde{H}(e) + \sum_{f\in E_{t+1}^T | R(e)\subseteq R(f)} y_f - y_e \qquad \forall 1 < t < T_{\max},\ e \in E_t^T, \tag{18}$$

$$x(e) \geqslant \tilde{H}(e) - y_e \qquad \forall e \in E_{T\max}^T, \tag{19}$$

$$x(e) \in \mathscr{C} \qquad \forall e \in E^T, \tag{20}$$

$$y_e \in \mathbb{N} \qquad \forall 1 < t \leqslant T_{\max}, e \in E_t^T. \tag{21}$$

If we enforce that all $y_e = 0$, then all capacities $x(e)$ must suffice to transport all travellers using only the type edges in their best path. The model, however, can decide to use different type edges (still part of the prescribed path) through the use of the variables y_e. These y_e variables model the number of people that were assigned to use type edge e of type t, but instead will be assigned to underlying type edges of type $t-1$. Note that in this way, these travellers may then again be reassigned to edges of type $t-2$, etc. The structure of the constraints for the type edges depends on the types of the edges. For an edge e of type $t=1$ in (17), there are no possibilities for rerouting passengers through y_e, since there are no edges of lower type. A similar argument for edges of type $T_{\max}$ in (19) makes it clear that we can only reassign passengers from these edges, not to them.

Next, we will prove the equivalence between MCF and IP_{XY}. To do so, let us first make the following observations concerning feasible flows.

Observation 4.12. For an arbitrary commodity k and an arbitrary type edge $e \in E_t^T$ that is allowed for this commodity, i.e., with $R(e)\subseteq P_k$, exactly one of the following holds:

$$e \in P_k^T \qquad \text{or} \qquad \exists t' > t : \exists f \in E_{t'}^T : R(e) \subseteq R(f) \subseteq P_k.$$

Thus, either a type edge e is part of the best path for commodity k, or there exists a type edge f of higher type that can also be used by k at this part of his path.

Next, Observation 4.13 considers the sum of the demand for all commodities that are allowed to use some track edge e. It is easy to see that, for any feasible flow F, this is equal to the sum of all the flows across e, i.e., to the sum of the flows on the track edge e, and on all type edges f of a type $t>1$ for which $e\in R(f)$.

Observation 4.13. For every feasible flow F of MCF, the following holds for every edge e of type 1:

$$\tilde{H}(e) + \sum_{f\in E_{t>1}^T | e\subseteq R(f)} \tilde{H}(f) = \sum_{k | R(e)\subseteq P_k} F_e^k + \sum_{f\in E_{t>1}^T | e\subseteq R(f)} \sum_k F_f^k \qquad \forall e \in E_1^T.$$

Lemma 4.14. *Every solution (X,F) of MCF can be transformed into a solution (X, Y) of IP_{XY} with the same objective function value.*

Proof. We prove this lemma by induction on $T_{\max}$. First, consider the case in which $T_{\max}=1$. From Observation 4.13 we know that $\tilde{H}(e) = \sum_{k | R(e)\subseteq P_k} F_e^k$ for all type edges $e \in E_1^T$, and thus that

$$x(e) \geqslant \sum_k F_e^k = \sum_{k|R(e)\subseteq P_k} F_e^k = \tilde{H}(e) \qquad \forall e \in E_1^T.$$

As induction hypothesis, let us now assume that the lemma holds for $T_{\max} = t^*$, and consider the case with $T_{\max} = t^* + 1$. For any edge e of type $t^* + 1$ we construct $y_e = \tilde{H}(e) - \sum_k F_e^k$. Note that $e \in E_{t^*+1}^T$. Note that $e \in E_{t^*+1}^T$ implies that y_e is non-negative. Clearly, now

$$x(e) \geqslant \sum_k F_e^k = \tilde{H}(e) - y_e \qquad \forall e \in E_{t^*+1}^T.$$

The final step is to reduce the problem from $t^* + 1$ types to t^* types by removing all the type edges of type $t^* + 1$ and the associated variables from the problem. The original solution (X,F) is now also feasible for the MCF of the reduced problem with $T_{\max} = t^*$. Therefore, we can apply the induction hypothesis and thus prove this lemma. □

Lemma 4.15. *Every solution (X, Y) of IP_{XY} can be transformed into a solution (X, F) of MCF with the same objective function value.*

Proof. From Lemmas 4.2 and 4.3 it is clear that we should show that feasible flows can be constructed from (X, Y) for artificial commodities $k = \{v, w\}$ for type edges $\{v, w\} \in E^T$, with demand $\tilde{H}(k)$. We will prove this lemma using induction on $T_{\max}$. First, note that for $T_{\max} = 1$ all constraints of (17) are of the form $x(e) \geqslant \tilde{H}(e)$ for all $e \in E^T$. Since there is only one type of edges, we can thus set

$$F_e^k = \tilde{H}(e) \qquad \forall k \in E^T, \quad e \in P_k^T = \{k\}.$$

Thus, every commodity corresponds to an edge in E^T, and $F_f^k = 0$ for all $f \neq k$. Obviously, all flow restrictions (11)–(15) are satisfied.

Next, assume that we can construct feasible flows for $T_{\max} = t^*$. Now we show that it is also possible to construct feasible flows for $T_{\max} = t^* + 1$. The constraints (19) for the type edges $e \in E_{t^*+1}^T$ are

$$x(e) \geqslant \tilde{H}(e) - y(e) \qquad \forall e \in E_{t^*+1}^T.$$

The total flow across type edge e can thus be found by taking F_e^k such that

$$F_e^k = \tilde{H}(e) - y_e.$$

The remaining demand y_e will be routed along the other possible edges: the type edges $f \in E_{t^*}^T$ for which $R(f) \subseteq R(e)$. The last part of this proof is to show that we are now not only able to construct a feasible flow for the edges of type $t^* + 1$, but additionally, also for all other edges. To show that this is possible, note that by restriction (18) we know that

$$x(f) \geqslant (\tilde{H}(f) + y(e)) - y(f) \qquad \forall f \in E_{t^*}^T.$$

This implies that, by our induction hypothesis, we can also find a feasible flow for the remaining edges of types $t \leqslant t^*$. □

The previous two lemmas imply the following theorem.

Theorem 4.16. *The problems MCF and IP_{XY} are equivalent.*

Lemma 4.17. *Consider the relaxation of IP_{XY} in which all $y_e \in \mathbb{R}_+$. Every solution (X, Y) of this relaxation, where Y is the vector containing all y_e, can be transformed into an integer solution $(X, \tilde{Y})$ with the same objective function value.*

Proof. We show that setting all y_e to the rounded down value $\tilde{y}_e \equiv \lfloor y_e \rfloor$ results in a feasible solution $(X, \tilde{Y})$. First, consider this rounding scheme for type edges e of type $t = T_{\max}$. The integrality of $x(e)$ and $\tilde{H}(e)$ ensures that $x(e) \geqslant \tilde{H}(e) - \lfloor y_e \rfloor$. Since $\tilde{y}_e \leqslant y_e$, all other restrictions also remain satisfied. Next, consider the edges of type $t = T_{\max} - 1$. Clearly, in (18) the sum of all $\tilde{y}_f$ of the type edges $f \in E^T_{t+1} | R(e) \subseteq R(f)$ is integer. Therefore, by applying similar reasoning as before, we see that the rounding scheme preserves the feasibility of the constructed solution for the remaining edges of types $T_{\max} - 1$ through $t = 1$. □

4.3. The integer programming formulation (IP_X)

In the IP_{XY} model, we introduced additional y_e variables to model the rerouting of commodities over other edges in the type graph. In this section, we describe an alternative model. This model does not use the rerouting variables y_e, but instead guides the routing by imposing additional restrictions.

Let us first review an example of this model for $T_{\max} = 2$.

Example 4.18. Consider the network displayed in Fig. 7. For every edge in the track graph, we consider all combinations of type edges that can be used to cross this edge. In this light, the following constraints are necessary, and, as we show later, also sufficient to model ECP. For the track edge e, and the crossing type edges f_1 and f_2:

$$
\begin{aligned}
&x(e) \geqslant \tilde{H}(e),\\
&x(e) + x(f_1) \geqslant \tilde{H}(e) + \tilde{H}(f_1),\\
&x(e) + x(f_1) + x(f_2) \geqslant \tilde{H}(e) + \tilde{H}(f_1) + \tilde{H}(f_2),\\
&x(e) + x(f_2) \geqslant \tilde{H}(e) + \tilde{H}(f_2).
\end{aligned}
$$

For the track edge g, and the crossing type edge f_1:

$$
\begin{aligned}
&x(g) \geqslant \tilde{H}(g),\\
&x(g) + x(f_1) \geqslant \tilde{H}(g) + \tilde{H}(f_1).
\end{aligned}
$$

For the track edge h, and the crossing type edge f_2:

$$
\begin{aligned}
&x(h) \geqslant \tilde{H}(h),\\
&x(h) + x(f_2) \geqslant \tilde{H}(h) + \tilde{H}(f_2).
\end{aligned}
$$

For example, the restriction $x(g) + x(f_1) \geqslant \tilde{H}(g) + \tilde{H}(f_1)$ enforces that the combined capacity of the type edges g and f_1 must suffice to transport all commodities that can at best use g, plus all that can at best use f_1. Since the only possibility to arrive at c is to use either g or f_1, it is clear that this is a necessary restriction.

Fig. 7. The track graph (a) and the type graph (b).

The integer programming formulation I_{PX} for ECP reads as follows:

$$\min \quad \sum_{e\in E^T} f(x(e)) \tag{22}$$

$$\text{s.t.} \quad x(e) + \sum_{t>1}\sum_{f\in S_t^e} x(f) \geqslant \tilde{H}(e) + \sum_{t>1}\sum_{f\in S_t^e} \tilde{H}(f) \qquad \forall e \in E_1^T, \quad \forall S_2^e, \ldots, \forall S_{T_{\max}}^e, \tag{23}$$

$$x(e) \in \mathscr{C} \qquad \forall e \in E^T, \tag{24}$$

where for a given edge e of type 1 the sets S_t^e are such that $S_t^e \subseteq \bigcup_{f\in S_{t-1}^e}\{g \in E_t^T | R(f) \subseteq R(g)\}$ with $S_1^e = \{e\}$. Thus, S_2^e is a subset of all the type 2 edges that cover e. Next, S_3^e is then a subset of all the type 3 edges that cover some edge in the current S_2^e, etc. Restriction (23) enforces sufficient capacity on type edge e together with the type edges in the sets S_t^e for $t=2,\ldots,T_{\max}$. Note that a constraint is added for all possible sets $S_2^e, \ldots, S_{T\max}^e$. For ease of reference, we refer to these restrictions as the *capacity-subset* (CS) constraints. Finally, feasible values for $x(e)$ are enforced by the set $\mathscr{C} \subset \mathbb{N}$, which represents the valid capacities that can be assigned to a type edge.

We will now show that the edge capacities X of any feasible solution of IP_{XY} are also a feasible solution for IP_X.

Lemma 4.19. *Every solution (X,Y) of IP_{XY} can be transformed into a solution X for IP_X with the same objective function value.*

Proof. We will show that all restrictions of IP_X are valid for IP_{XY}. Consider an arbitrary restriction of IP_X for type edge $e \in E_1^T$, and with the sets $S_1^e, \ldots, S_{T\max}^e$. Recall the CS restriction (23) for type edge e:

$$x(e) \geqslant \tilde{H}(e) + \sum_{f\in E_2^T | R(e)\subseteq R(f)} y_f.$$

Now, consider an arbitrary collection $S_2^e, \ldots, S_{T\max}^e$ of subsets, i.e., an arbitrary CS constraint. Since all values y_e are non-negative, the following holds:

$$\begin{aligned} x(e) + \sum_{t>1}\sum_{f\in S_t^e} x(f) &\geqslant \tilde{H}(e) + \sum_{f\in E_2^T | R(e)\subseteq R(f)} y_f + \sum_{t>1}\sum_{f\in S_t^e} x(f) \\ &\geqslant \tilde{H}(e) + \sum_{f\in S_2^e | R(e)\subseteq R(f)} y_f + \sum_{t>1}\sum_{f\in S_t^e}\left(\tilde{H}(f) + \sum_{g\in S_{t+1}^e | R(f)\subseteq R(g)} y_g - y_f\right)\cdots \\ &\geqslant \tilde{H}(e) + \sum_{t>1}\sum_{f\in S_t^e}\tilde{H}(f). \end{aligned}$$

To prove the last step we need to show that the y_e variables with a negative sign cancel out against the other y_e variables. This is shown as follows:

$$\{f \in S_2^e | R(e) \subseteq R(f)\} \cup \bigcup_{\substack{t>1 \\ f\in S_t^e}} \{g \in S_{t+1}^e | R(f) \subseteq R(g)\} = \{f \in S_2^e\} \cup \bigcup_{\substack{t>1 \\ f\in S_t^e}} \{g \in S_{t+1}^e\} \subseteq \bigcup_{t>1}\{f \in S_t^e\}.$$

The equality holds because, by definition, every $g \in S_{t+1}^e$ has *some* $f \in S_t^e$ for which $R(f) \subseteq R(g)$. Since we take the union over *all* $f \in S_t^e$, the equality follows immediately. This completes the proof. □

The inverse, extending a solution X of IP_X with appropriately chosen values for Y, gives a feasible solution to IP_{XY}, as is shown in Lemma 4.20.

Lemma 4.20. *Every solution X of IP_X can be transformed into a solution (X,Y) for IP_{XY} with the same objective function value.*

Proof. As in the proof of Lemma 4.14, we will again provide a scheme for constructing a suitable vector Y. Consider an arbitrary solution X of IP_X. Let us choose the value of y_e for any type edge $e \in E_t^T$ as

$$y_e = \left(\tilde{H}(e) - x(e) + \sum_{f \in E_{t+1}^T : R(e) \subseteq R(f)} y_f \right)^+, \tag{25}$$

where $(x)^+ \equiv \max(0,x)$. Recursively, we can thus construct all y_e starting at type edges e of type $t = T_{\max}$ (for which $E_{t+1}^T = \emptyset$), and ending at $e \in E_2^T$. We are now left to prove that

$$x(e) \geqslant \tilde{H}(e) + \sum_{f \in E_2^T | R(e) \subseteq R(f)} y_f \qquad \forall e \in E_1^T.$$

Thus, substituting (25) for all y_f, we have to prove the validity of

$$x(e) \geqslant \tilde{H}(e) + \sum_{f \in E_2^T | R(e) \subseteq R(f)} \left(\tilde{H}(f) - x(f) + \sum_{g \in E_3^T | R(f) \subseteq R(g)} (\tilde{H}(g) - x(g) + \cdots)^+ \right)^+.$$

To show this, consider the values of the different max-plus parts, i.e., the y_f. Being either zero or positive, we introduce the sets $S_t^* \subseteq E_t^T$ such that $S_t^* = \{f \in E_t^T | y_f > 0\}$ for our arbitrary solution X. Since the capacity restrictions (23) of the IP_X formulation contain all possible combinations of sets S_t^e, we know that all sets S_t^* are among them. Therefore

$$\begin{aligned}
&\tilde{H}(e) + \sum_{f \in E_2^T | R(e) \subseteq R(f)} \left(\tilde{H}(f) - x(f) + \sum_{g \in E_3^T | R(f) \subseteq R(g)} (\tilde{H}(g) - x(g) + \cdots)^+ \right)^+ \\
&\quad = \tilde{H}(e) + \sum_{f \in S_2^*} \left(\tilde{H}(f) - x(f) + \sum_{g \in S_3^*} (\tilde{H}(g) - x(g) + \cdots) \right) \\
&\quad = \tilde{H}(e) + \sum_{f \in S_2^*} (\tilde{H}(f) - x(f)) + \sum_{g \in S_3^*} (\tilde{H}(g) - x(g)) + \cdots \\
&\quad = \tilde{H}(e) + \sum_{t>1} \sum_{f \in S_t^*} \tilde{H}(f) - \sum_{t>1} \sum_{f \in S_t^*} x(f) \leqslant x(e).
\end{aligned}$$

As in the proof of Lemma 4.14, replacing the nested summations by the separate summations in the second equation can be done using the results from Lemma 4.10. Since this construction is valid for an arbitrary solution X, this completes the proof. □

The previous two lemmas imply the following theorem.

Theorem 4.21. *The problems IP_X and IP_{XY} are equivalent.*

The number of CS restrictions (23) is exponential in the number of type edges. However, as we will show now, we can find the maximally violated CS constraint for every edge e of type 1 in polynomial time.

To solve this separation problem, consider an edge e of type 1, and construct the directed graph $T(e)$ that contains a node for every type edge that covers e, i.e., a node for every $f \in E^T$ with $e \in R(f)$. The nodes can be layered according to the types of the associated edges. The graph contains only arcs between nodes of layer t to layer $t-1$. Let us say that the node in layer t is related to type edge $g \in E_t^T$, while the node in layer $t-1$

is related to type edge $f \in E^T_{t-1}$. There exists an arc between these two nodes if the type edge f is the unique type edge (according to Lemma 4.10) for which $e \subseteq R(f) \subseteq R(g)$. Thus, every node in $T(e)$ has exactly one outgoing arc, though possibly many incoming arcs. It is easy to see that $T(e)$ is a directed in-tree rooted at the node associated with the type edge e.

Next, consider a solution $\tilde{X}$ with a value $\tilde{x}(e)$ for every $e \in E^T$. For simplicity, let us refer to the node in $T(e)$ that is associated with type edge f as node f, etc. With every node f in $T(e)$ we associate a revenue (violation) $\tilde{H}(f) - \tilde{x}(f)$. The problem is to find a subtree of $T(e)$ that is rooted at e for which the sum of the revenues of the nodes in the subtree is maximal. Such a subtree that has a positive revenue, corresponds to a violated CS inequality.

Lemma 4.22. *Consider a layered tree $T(e)$ for an edge e of type 1. Every subtree of $T(e)$ that is rooted at e corresponds to a CS inequality of (23) for type edge e and with the set S^e_t equal to the nodes in the subtree that are in layer t, for $t \in \{2, \ldots, T_{max}\}$.*

Proof. We have to show that

$$S^e_t \subseteq \bigcup_{f \in S^e_{t-1}} \{g \in E^T_t | R(f) \subseteq R(g)\}$$

with $S^e_1 = \{e\}$ holds. By the construction of $T(e)$, and the fact that the subtree is connected, this result follows immediately. □

To solve the problem of finding the subtree that maximises the total node revenue we add a label $d(f)$ to every node f, and initialise it to $d(f) = \tilde{H}(f) - \tilde{x}(f)$.

Algorithm 1. Determine the maximally violated CS constraint on type edge e.

Input: A directed in-tree $T(e)$ that is layered, and rooted at node e.
A solution $\tilde{x}$.
Initialise $d(f) = \tilde{H}(f) - \tilde{x}(f)$ for each node f in $T(e)$.
Initialise $t = t^*$, where t^* is the highest type for which there are type edges represented in $T(e)$.
While $t > 1$ **do**
  For all nodes g in layer t **do**
    If $d(g) > 0$ **then**
      Let f be the unique node in layer $t-1$ to which g is connected.
      Set $d(f) \leftarrow d(f) + d(g)$
    else
      Delete the subtree rooted at g from $T(e)$.
    end if
  end for
  Set $t \leftarrow t - 1$.
end while

In Algorithm 1, we examine the tree $T(e)$ starting at the nodes in layer t^*, where t^* is the highest type for which there are type edges represented in $T(e)$. Note that t^* can be strictly smaller than $T_{\max}$. For every node g in layer t that has a positive label $d(g)$, let f be the unique node in layer $t-1$ to which g is connected. For these nodes g with $d(g) > 0$, set $d(f) \leftarrow d(f) + d(g)$. Otherwise, if $d(g) \leqslant 0$, delete the subtree rooted at g from $T(e)$. Note that the remaining graph $T(e)$ is still a directed in-tree rooted at e. Now, set $t \leftarrow t - 1$, and proceed with the next layer as before.

After $t-1$ iterations, we arrive at layer 1 that contains only node e. All that is left of $T(e)$ is the subtree of the original tree, rooted at e that has a total revenue of $d(e)$. Note that the running time of the algorithm is $\mathcal{O}(T_{\max}|E^T|)$, and therefore polynomial in the size of the input.

Lemma 4.23. *When processing layer t in Algorithm 1, the value of d(f) for any node at layer t is the maximum revenue of any subtree of the original T(e) that is rooted at f.*

Proof. We show this by induction on the current layer t. For the initial layer at $t=t^*$, the value $d(f)=\tilde{H}(f)-\tilde{x}(f)$ is exactly equal to the total revenue of the subtree rooted at f, i.e., only node f. Now, assuming that, at some layer t, the claim holds. For an arbitrary node f of layer $t-1$, the algorithm takes all, and only those subtrees rooted at nodes g in layer t with $d(g)>0$. Clearly, if $d(g)\leqslant 0$, then the total revenue would become no better than if the subtree rooted at g were deleted. On the other hand, if a subtree with $d(g)>0$ were deleted, then this would strictly worsen the overall revenue of the subtree rooted at f. Therefore, $d(f)$ is also the maximal revenue of any subtree rooted at f for any node f in layer $t-1$. This completes the proof. □

Corollary 4.24. *The remaining subtree is the maximal subtree of the original tree T(e) that is rooted at e, with respect to the total revenue over all nodes.*

Proof. This follows immediately from the previous lemma. □

If $d(e)$ is positive, then, according to Corrollary 4.24 and Lemma 4.22 we have found a violated CS constraint. Alternatively, if $d(e)\leqslant 0$, then all CS constraints for the edge e of type 1 are satisfied.

Theorem 4.25. *The separation problem for the exponentially many capacity-subset constraints of (23) can be solved in polynomial time in the size of the input.*

Proof. The separation problem for the CS constraints for one edge e of type 1 can be solved in polynomial time. Since we only have to test this for every edge of type 1, we can thus solve the overall separation problem for the CS constraints in polynomial time. □

5. Computational results

The three formulations MCF, IP_{XY}, and IP_X differ with respect to the necessary number of variables and constraints for ECP. Table 1 shows worst-case statistics for all formulations.

We have used three real-life instances of MLPP to compare the practical performance of solving them using the IP_{XY} and IP_X formulations, after applying the substitutions mentioned in Section 3. The instances all concern different parts of the Dutch railway network. We have chosen these instances because of their different structures of the associated track graphs. The characteristics of the instances can be found in Table 2.

Table 1
Variable and constraint statistics for the MCF, IP_{XY} and IP_X models of ECP

	MCF	IP_{XY}	IP_X
# Vars.	$\lvert E^T\rvert+\mathcal{O}(\lvert E^T\rvert^2)$	$\lvert E^T\rvert+\lvert E^T\setminus ET_1\rvert$	$\lvert E^T\rvert$
# Cons.	$\lvert E^T\rvert+\mathcal{O}(\lvert E^T\rvert\lvert V\rvert)$	$\lvert E^T\rvert$	$\mathcal{O}(2^{\lvert E^T\rvert})$

Table 2
Instance characteristics

Instance	$T_{\max}$	$\|L\|$	$\|V\|$	$\|E\|$	$\|E^T\|$
NS3600	3	64	28	27	50
NSNH	3	81	36	37	58
NSRandstad	3	331	122	138	204

The three networks are visualised in the graphs in Fig. 8. The first two instances are rather small with respect to the numbers of nodes in the network. We have chosen these instances to compare the practical use of the two proposed models. At first glance, one might expect that there will be a computational trade-off between the model structure of IP_{XY} and the additional CS constraints in IP_X. The track graph of

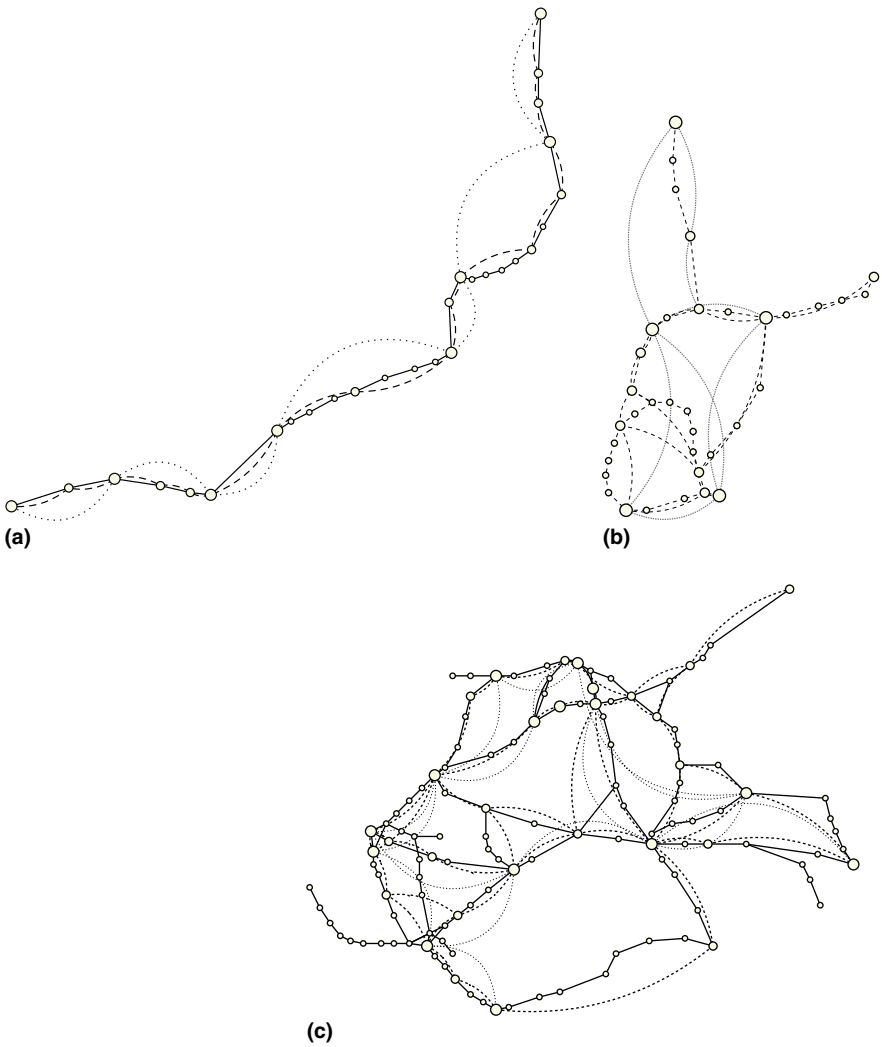

Fig. 8. The type graphs for the instances NS3600 (a), NSNH (b) and NSRandstad (c).

NS3600 is a path. This influences the number of type edges in E^T. From Lemma 4.7, it is easy to see that, in the case of a path, the number of edges in E^T is at most $T_{\max} \cdot |E|$. But even more important, the number of possible subsets for IP_X is also at most $T_{\max} \cdot |E|$, i.e., at most $T_{\max}$ per track edge $e \in E$. To test the behaviour of the number of subset restrictions, we have also included two instances that introduce stations with a degree higher than 2 in the track graphs.

Every instance was tested using both the IP_{XY} and the IP_X formulation. The numerical results were obtained using CPLEX 7.5 on an AMD Athlon 800 MHz with 512 MB internal memory running Linux, kernel 2.4.8. All instances where tested with all the CPLEX parameters at their default values. The model statistics and computational results are shown in Tables 3 and 4 respectively.

The most striking statistics of Table 3 are the number of CS constraints for IP_X. The table shows that, for these real-life instances, the structure and size of the networks call for a number of subsets that is roughly only linear in the number of type edges $|E^T|$. Therefore, we have not implemented the separation algorithm described in Section 4.3, but simply added all CS constraints at the root node.

The computational results in Table 4 show that none of the IP_{XY} instances could be solved to optimality within one hour. One possible explanation for this could be the significantly lower root LP values. With NS3600 for example, the root LP value of the IP_X formulation is 7213 (4.08% gap), whereas one hour, or 1.2 million nodes, of branching on IP_{XY} gives a best lower bound of only 7173 (4.61% gap). Similar con-

Table 3
Statistics for the different instances and models

Instance	Model	# var.	# con.	# subsets
NS3600	IP_{XY}	1303	114	–
NS3600	IP_X	1280	145	81
NSNH	IP_{XY}	1641	139	–
SNH	IP_X	1620	230	149
NSRandstad	IP_{XY}	6686	535	–
NSRandstad	IP_X	6620	734	403

Table 4
Computational results

Instance	Model	Best	Root LP	Best LP	Gap (%)	# sec.	# nodes
NS3600	IP_{XY}	7520	6430	7173	4.61	†	1206968
NS3600	IP_X	7520	7213	7520	0	0.81	60
NSNH	IP_{XY}	13760	12501	13313	3.25	†	747874
NSNH	IP_X	13760	13133	13760	0	6.66	407
NSRandstad	IP_{XY}	55360	46146	48733	11.97	†	104234
NSRandstad	IP_X	52480	48880	50510	3.75	†	25008

The dagger (†) indicates that the time limit of 3600 seconds was reached.

Table 5
A comparison between the MLPP solutions, and solutions of using a system split

Instance	MLPP	LPP	Reduction (%)
NS3600	7520	9440	−20.3
NSNH	13760	17600	−21.8
NSRandstad	52480	62240	−15.7

clusions can also be drawn from the other results. It can be concluded that the results for IP_X are more promising than those for IP_{XY}.

We have also tested the effect of using a system split and solving a separate line planning problem for every type, compared to solving the line planning problem using MLPP. The system split was made according to the best-paths for every commodity. Thus, the capacities must allow all passengers to use lines of their best (highest) possible type. Using such a system split is equivalent to forcing all y_e variables in an IP_{XY} instance to zero. The results for this system split are shown in Table 5. The reduction in the objective function value (the total amount of operated capacity) by using MLPP is between 15.7% and 21.8% for the tested instances. Note once more that these models only minimise the operated capacities and operational costs, and do not measure, for example, the necessary number of train changes of passengers that travel through the network.

6. Summary and conclusions

In this paper we have described different integer programming formulations for modelling the MLPP. Where previous work, e.g., Bussieck [2], Claessens et al. [4], focused on modelling LPP for exactly one type of trains and stations, we present generalisations of these models within a cost-minimising setting. First, the general multi-commodity flow formulation is introduced in Section 4.1. This formulation is then used to prove the validity of the two main formulations IP_{XY} (Section 4.2) and IP_X (Section 4.3). Using three real-life instances we compare the computational results for both formulations. From these tests, we can conclude that the IP_X formulation outperforms IP_{XY} in all of the chosen instances. Even though the number of restrictions of IP_X can be exponential in the size of the instance, we show in Section 4.3 how to identify violated constraints in polynomial time.

Future research on the topic of multi-type line planning problems will focus on using techniques such as branch-and-cut to solve even larger instances. In addition, we will consider new classes of model restrictions, e.g., track or station utilisation constraints, aiming at improving the practical applicability of the solutions.

Acknowledgement

This research is partly sponsored by the Human Potential Program of the European Union under contract no. HPRN-CT-1999-00104 (AMORE).

References

[1] A. Alfieri, R. Groot, L.G. Kroon, A. Schrijver, Efficient Circulation of Railway Rolling Stock. Technical Report ERS-2002-110-LIS, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam, Rotterdam, The Netherlands, 2002.
[2] M.R. Bussieck, Optimal lines in public rail transport, PhD thesis, Technical University Braunschweig, Braunschweig, Germany, 1998.
[3] A. Caprara, M. Fischetti, P. Toth, D. Vigo, P.L. Guida, Algorithms for railway crew management, Mathematical Programming 79 (1997) 125–141.
[4] M.T. Claessens, N.M. van Dijk, P.J. Zwaneveld, Cost optimal allocation of rail passenger lines, European Journal of Operational Research 110 (1998) 474–489.
[5] R. Freling, R.M. Lentink, L.G. Kroon, D. Huisman, Shunting of passenger train units in a railway station. Technical Report EI2002-26, Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands, Transportation Science, in press.
[6] G. Gallo, F. Di Miele, Dispatching busses in parking depots, Transportation Science 79 (2001) 322–330.

[7] J.H.M. Goossens, C.P.M. van Hoesel, L.G. Kroon, A branch-and-cut approach for solving line planning problems, METEOR Research Memorandum RM/01/016, University of Maastricht, Maastricht, The Netherlands, Transportation Science, in press.
[8] L.G. Kroon, M. Fischetti, Crew Scheduling for Netherlands Railways "Destination: Customer", in: Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, vol. 505, Springer-Verlag, Berlin, 2001.
[9] K. Nachtigall, Periodic Network Optimization and Fixed Interval Timetables, Habilitation thesis, Deutsches Zentrum für Luft- und Raumfahrt e.V., Braunschweig, Germany, 1999.
[10] M.A. Odijk, Railway Timetable Generation, PhD thesis, Delft University of Technology, Delft, The Netherlands, 1997.
[11] C. Oltrogge, Linienplanung für mehrstufige Bedienungssysteme im öffentlichen Personenverkehr, PhD thesis, Technical University Braunschweig, Braunschweig, Germany, 1994 (in German).
[12] M. Peeters, L.G. Kroon, Circulation of railway rolling stock: A branch-and-price approach, Technical report, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam, Rotterdam, The Netherlands, 2003.
[13] A. Schrijver, Minimum circulation of railway stock, CWI Quarterly 3 (1993) 205–217.
[14] A. Schrijver, A. Steenbeek, Dienstregelingontwikkeling voor Railned (Timetable construction for Railned), Technical report, CWI, Amsterdam, The Netherlands, 1994 (in Dutch).
[15] T. Winter, U.T. Zimmermann, Real-time dispatch of trams in storage yards, Annals of Operations Research 96 (2000) 287–294.
[16] P.J. Zwaneveld, L.G. Kroon, C.P.M. van Hoesel, Routing trains through a railway station based on a node packing model, European Journal of Operational Research 128 (1) (2001) 14–33.