

On solving multi-type line planning problems

Citation for published version (APA):

Goossens, J. H. M., van Hoesel, C. P. M., & Kroon, L. G. (2002). *On solving multi-type line planning problems*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 009 <https://doi.org/10.26481/umamet.2002009>

Document status and date:

Published: 01/01/2002

DOI:

[10.26481/umamet.2002009](https://doi.org/10.26481/umamet.2002009)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

On solving multi-type line planning problems

Jan-Willem Goossens^{*} Stan van Hoesel[†] Leo Kroon[‡]

February 28, 2002

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; Branch and cut; Combinatorial optimization; Railway transportation

1 Introduction

The planning problem faced by every railway operator consists of several consecutive stages, ranging from strategic decisions to operational traffic control. Strategic problems are driven by estimates for the long-term demand. The first problem concerns the determination of the infrastructure, such as railway tracks and stations. Both the infrastructure and demand data are input for the 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 train capacity is enough to meet the passenger demand. Successive decision stages are the more detailed planning problems such as the construction of timetables [SS94, Odi97, Nac99], traffic planning (route assignment, platform assignment [Zwa97]), rolling stock planning [Sch93], personnel planning [CFT⁺97], and shunting planning [DMG98].

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* (of lines). The line planning models described in the literature have thus far considered all lines to be of the same type [Bus98, CDZ98, GHK01]. 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

^{*}Dept of Quantitative Economics, University of Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands. E-mail: j.goossens@ke.unimaas.nl

[†]Dept of Quantitative Economics, University of Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands. E-mail: s.vanhoesel@ke.unimaas.nl

[‡]Rotterdam School of Management, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands. E-mail: l.kroon@fbk.eur.nl

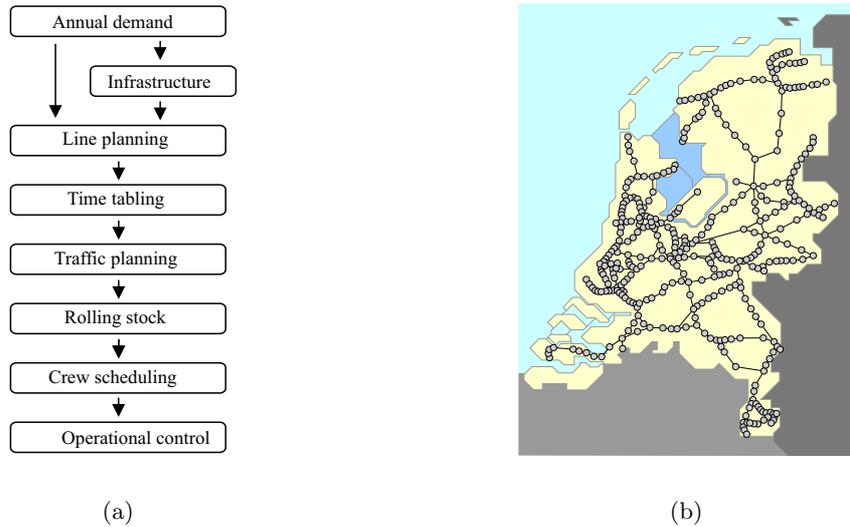


Figure 1: The different stages in the planning process (1(a)) and the Dutch railway network (1(b))

separate problems for every type. This allocation was determined for example by the procedure System Split [Olt94]. In this paper, we introduce several generalizations of the previous models for simultaneously solving line planning problems with multiple train types. An evaluation using real life instances is included.

Traditionally, the objective when constructing a line plan has been to find a set of lines that maximizes the number of direct travellers, i.e. the number of travellers that do not have to change trains during their journey, cf. [Bus98]. This is an obvious objective from a service perspective. However, this objective tends to generate geographically long train lines. As an alternative objective, similar to [CDZ98], [Bus98] and [GHK01], this paper focuses on models for minimizing the operational costs of a line plan.

The next section considers the model formulations. Apart from a multi-commodity flow formulation, we develop two alternative mathematical models and prove their equivalence. In Section 3 we describe the preliminary computational study, based on instances of the Dutch railway operator NS Reizigers.

2 Assumptions and notation

As described in [GHK01], the concept of a line is fundamental in railway systems. A line specifies a route between an origin and a destination station and the subsequent stops, combined with an operated hourly frequency. A line plan is the set of operated lines. The line plan does not incorporate the exact time table for the operated lines, though we assume that the time table 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 minimizes the induced operational costs [CDZ98, Bus98, GHK01].

Let us consider the track graph $G = (V, E)$ built up of the set of vertices (stations) V

with $|V| = n$, and connecting edges (tracks) E . For this track graph, 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 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 and can thus be fixed a priori. The assumption that there is exactly one fixed route is not important. The essence is that the route is known for every commodity. The demand of commodity k , i.e. the number of travellers that want to travel from s^k to t^k , is given by its entry in the $n \times n$ demand matrix $H: H^{s^k t^k}$, also denoted by H^k . We will assume that this matrix is symmetric, i.e. that it has 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 $T = \{1, \dots, T_{\max}\}$, 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 categorization 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 $t_v \geq t$.

Example 2.1

Consider the track graph in Figure 2. 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. Train line 1 of type 1, going from station a to station w , halting at all stations in between. Train 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 stations, 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 route of the train lines.

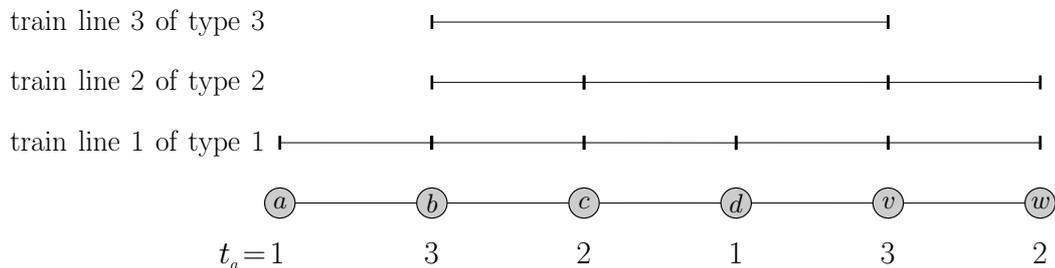


Figure 2: The track graph G showing the types of the stations and several train lines.

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 $T_{\max} - 1$ new sets of edges. See Example 2.2.

Example 2.2

The track graph G given in Figure 2 can be transformed into the type graph $G^T = (V, E^T)$ displayed in Figure 3. Note that the structure of the type graph depends on the types of the stations, not on the lines.

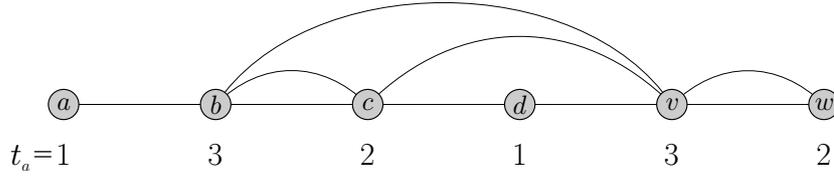


Figure 3: The type graph, based on the network from Figure 2.

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

Let us make the following assumptions:

- The track set is equal to the lowest set of type tracks, i.e. that E is equal to E_1^T .
- The route definitions are consistent in that if the route $R(e)$ of edge $e = \{i, j\} \in E^T$ contains two vertices v and w for which $f = \{v, w\} \in E^T$, then also $R(f) \subseteq R(e)$.
- 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 : $e \subseteq R(f) \subseteq R(g)$.

The track and type graphs in the previous two examples meet all of these assumptions. That $E = E_1^T$ can be seen immediately, even though we only show the overall E^T . The second assumption is for example illustrated by the type 3 edge $e = \{b, v\}$ and the type 3 edge $f = \{c, v\}$. For assumption three, note that 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, using $g = \{b, v\}$ we assume the presence of the type 2 edges $\{b, c\}$ and $\{c, v\}$. All of these assumptions are not very restrictive.

Note that the sets E_t^T need not be exhaustive, i.e. not every pair of vertices $v, w \in V_t$ for which there exists a simple path has to be present in E_t^T . Also note that G^T can be a multi graph, but only in the sense that some edge $\{v, w\}$ can be in E_t^T and $\{v, w\} \in E_{t'}^T$ for $t \neq t'$ (see the edges $\{b, c\}$ and $\{v, w\}$ in Example 2.2). 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 sets describe 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

$$e \in P_k^T \iff \begin{array}{l} R(e) \subseteq P_k \text{ and} \\ \nexists f \in E_{t'}^T : R(e) \subseteq R(f) \subseteq P_k \end{array} \quad (1)$$

Hence, in the graph in Figure 3 the best-edge path for $k = (a, w)$ is $P_{aw}^T = \{\{a, b\}, \{b, v\}, \{v, w\}\}$ where $\{a, b\} \in E_1^T$, $\{b, v\} \in E_3^T$ and $\{v, w\} \in E_2^T$.

2.1 The Edge Capacity and Line Planning problems

The Edge Capacity problem (ECP) is described on the track graph G and the associated type graph G^T . In short, the problem in ECP is to provide enough capacity on the edges in G^T such that all commodities can be transported simultaneously, while minimizing some objective function of the used capacity. The line planning problem (LPP) is a generalization of ECP. Where in ECP the decision variables model the supplied capacities for individual edges, in LPP these decisions are made for sets of edges (the lines).

If a line is selected to be part of the line plan at an hourly capacity of e.g. c , then it thus provides a capacity of c along all the edges in the type graph G^T that it uses. On every type edge in the network, the combined capacity of all the lines that use this edge should suffice to transport all the flow of the passengers along this edge.

The lines in a line plan are assumed to be operated with an identical capacity in both directions. This assumption is widely adopted by many authors (see [CDZ98, Bus98]). Consequently, the provided capacity on some edge $\{v, w\}$ of a line is the same in both directions. Note that there is flow across any edge $e = \{v, w\} \in E^T$ in two directions: flow in the direction from v to w and in the direction from w to v . Thus, the combined capacity of all lines using edge e should be enough such that the capacity on every edge is at least as large as the flow in each direction on that edge. Consider a network with two vertices and only one edge: $V = \{v, w\}$ and $E = E^T = \{v, w\}$. If 50 people want to travel from v to w , and 60 from w to v , then the combined capacity of the lines that are operated on $\{v, w\}$ should be at least $\max\{50, 60\} = 60$. The LPP is a generalization of the Edge Capacity problem. In ECP, the supplied capacity is set for individual edges, whereas in LPP this is done per set of edges (the lines). Only after discussing several models for describing ECP will we switch back to LPP by replacing the edge capacity variables $x(e)$ by the total sum of the capacity of the chosen lines on edge e . For the details, see Section 2.5.

2.2 The multi-commodity flow formulation (MCF)

Let us introduce two directed graphs, similar to the network and 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 backward arc (j, i) for every 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 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 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

$$\sum_{j:(i,j) \in A^T} F_{ij}^k - \sum_{j:(j,i) \in A^T} F_{ji}^k = b_i^k \quad \forall i \in V, \forall k \in V \times V \quad (2)$$

where the flow variables F_{ij}^k represent the number of passengers of the OD pair k that use arc (i, j) in their route. 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:

$$(3)$$

$$\min \quad \sum_{e \in E^T} f(x(e)) \quad (4)$$

$$\text{s.t.} \quad x(e) \geq \sum_k F_{ij}^k \quad \forall t \in T, \forall (i, j) \in A_t^T, e = \{i, j\} \in E_t^T \quad (5)$$

$$\sum_{j|(i,j) \in A^T} F_{ij}^k - \sum_{j|(j,i) \in A^T} F_{ji}^k = b_i^k \quad \forall i \in V, \forall k \in V \times V \quad (6)$$

$$F_{ij}^k = 0 \quad \forall k \in V \times V, \forall (i, j) \in A^T : \vec{R}((i, j)) \not\subseteq \vec{P}_{s^k t^k} \quad (7)$$

$$F_{ij}^k \in \mathbb{N} \quad \forall k \in V \times V, \forall (i, j) \in A^T \quad (8)$$

$$x(e) \in \mathcal{C} \quad \forall e \in E^T \quad (9)$$

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 (5) to enforce that the capacity of edge e , $x(e)$, is at least equal to the maximum of the total flow across both related arcs.

The restrictions of type (6) are the flow conservation constraints for every vertex. Restriction (7) enforces 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 $\mathcal{C} \subset \mathbb{N}$, 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 2.3

Let us preview the commodity decomposition principle on the track graph G and type graph G^T used in Example 2.2. Figure 2.3 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 2.1 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.

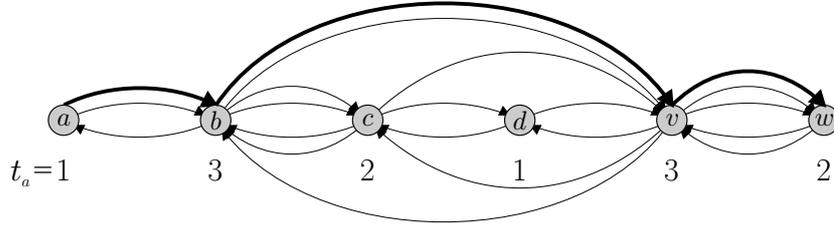


Figure 4: The digraph D^T based on the network in Figure 3.

Lemma 2.1 (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 $\vec{P}_{k_f}^T = \{f\}$. The reverse, combining of the flows, also again results in a feasible flow for k .

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

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

and vice versa, will satisfy 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 $f = (n, m) \in \vec{P}_{k_f}^T$:

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

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. \square

The application of the decomposition part of Lemma 2.1 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, reducing the total number of commodities in the system.

Lemma 2.2 (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 , and vice versa.

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 a red, and H^{k_2} of them blue. Clearly, these flows are still feasible flows. The reverse is shown by removing all the labels from both commodities. \square

By the previous two lemmas, we can assume that all commodities $k = (n, m)$ in an instance of MCF have the property that $\vec{P}_k^T = \{(n, m)\}$. For ease of notation, let us introduce $\tilde{H}(a) = \sum_{k|a \in \vec{P}_k^T} H^k$ for an arc $a \in A^T$. 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.

Example 2.4

Let us review the MCF problem on the graph displayed in Figure 3 on 4. Originally, this problem contained $6 \times 5 = 30$ 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 overall be reduced to 14.

We will use the previous two lemmas to show that we can assume that an optimal flow is symmetric. This is shown by using induction on the number of train types T_{\max} . The induction step, where we assume we can construct a symmetric solution for $T_{\max} = t^*$, we will start transforming the nonsymmetric flow across the arcs of type $T_{\max} = t^* + 1$.

Corollary 2.1

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 (1). Since arc (i, j) is of type T_{\max} , there do not exist any arcs $(i', j') \in A_{t > T_{\max}}^T$ with $(i, j) \neq (i', j')$. \square

Theorem 2.1 (Symmetric Flow)

Let us assume that the demand matrix H is symmetric. 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 will 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 the theorem holds for $T_{\max} = t^*$. We will show that this implies that it also holds for $T_{\max} = t^* + 1$. From Corollary 2.1 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 edge $\{i, j\} \in E_{t^*+1}^T$ equation (5) tells us

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

Suppose the two opposing flows defined for this edge are not symmetric. So, without loss of generality, assume $F_{ij}^{ij} < F_{ji}^{ji}$. From Lemma 2.2 we know that we can find $F_{ji}^{ji} - F_{ij}^{ij}$ units of flow of commodity (i, j) and 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 (5) 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 edges below $\{i, j\}$. The resulting flow is feasible for MCF, and is symmetric on all edges of type $t^* + 1$. 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 problem. Therefore, by the induction hypothesis, this MCF can be made symmetric. The overall cost will now be $\sum_{e \in E_{t^*+1}^T} f(x^*(e)) + \sum_{e \in E_{t \leq t^*}^T} f(x^*(e)) = \sum_{e \in E^T} f(x^*(e))$. \square

Because of the result of Theorem 2.1 we will no longer distinguish the commodities (n, m) and (m, n) , or the arcs (i, j) and (j, i) , for 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 type graph.

Lemma 2.3

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 edge f of type t for which $e \in R(f)$.

Proof. Without loss of generality, we can rename the vertices and edges of G such that $V = \{1, \dots, n\}$ and $E = \{\{v, v+1\} | v \in \{1, \dots, n-1\}\}$, since G is a path. The proof is by contradiction. Assume that for an arbitrary type t , there are two distinct 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 $v \leq i < w$ but also that either $w < j$ (crossing) or $j < w$ (non-crossing).¹ The first case implies that $t_w \geq t$ 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. \square

Corollary 2.2

For any two distinct edges f and g both of type t with $R(f) \cap R(g) \neq \emptyset$, we know that $R(f) \cup R(g)$ is not a path.

¹Note that if $w = j$, then we could reverse numbering of the vertices.

Proof. Assume that $R(f) \cup R(g)$ is a path. Thus, the graph induced by the edges in $R(f) \cup R(g)$ is a path, and since $R(f) \cap R(g) \neq \emptyset$, we know there is at least one edge e for which $e \in R(f)$ and $e \in R(g)$. This is not possible according to Lemma 2.3. \square

Corollary 2.3

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. This however, contradicts Corollary 2.2. \square

Lemma 2.4

Given some track edge $e \in E$. Now, for any type edge g of type t with $e \in R(g)$ there exists, for every type $t' < t$, 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 2.3. Moreover, since the existence and uniqueness also hold for this type edge of type t' , there thus also exists a unique type edge of type $t'' = t' - 1$ etc. \square

Put differently, there can not exist two overlapping edges $f, g \in E_t^T$ for which there exists a type edge h that covers both f and g , simply because the combined coverage of f and g can not be a path by Corollary 2.2.

2.3 The integer programming formulation (IP_{XY})

Solving ECP problems using the MCF formulation requires a large number of variables and restriction. 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 capacity of the edges in the type graph. For this, let us introduce additional variables y_e for every edge $e \in E_t^T$ with $t > 1$. They represent the number of travellers over all pairs (a, b) that could have used edge e across this particular part of their path P_{ab} , but don't. Capacity will be reserved for them on the underlying edges.

Example 2.5

Consider the type graph in Figure 5. 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 $y_{\{a,b\}}$ and y_e through which they could be assigned again 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

$$x(f) \geq \tilde{H}(f) - y_f$$

$$x(e) \geq \tilde{H}(e) + y_f - y_e$$

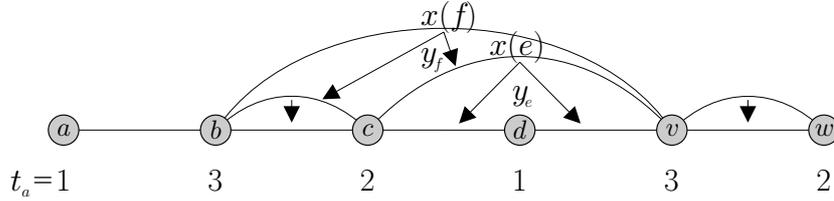


Figure 5: Travellers are possibly assigned to underlying edges.

The idea illustrated in this example can be generalized to the following model, referred to as IP_{XY} .

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

$$x(e) \geq \tilde{H}(e) + \sum_{f \in E_{t+1}^T | R(e) \subseteq R(f)} y_f \quad e \in E_1^T \tag{11}$$

$$x(e) \geq \tilde{H}(e) + \sum_{f \in E_{t+1}^T | R(e) \subseteq R(f)} y_f - y_e \quad 1 < t < T_{\max}, e \in E_t^T \tag{12}$$

$$x(e) \geq \tilde{H}(e) - y_e \quad e \in E_{T_{\max}}^T \tag{13}$$

$$x(e) \in \mathcal{C} \quad e \in E^T \tag{14}$$

$$y_e \in \mathbb{N} \quad 1 < t \leq T_{\max}, e \in E_t^T \tag{15}$$

Recall that $\tilde{H}(e)$ is the number of travellers for whom edge e is part of their best path P_k^T . Thus if we were to enforce that all $y_e = 0$, then all capacities $x(e)$ have to suffice to transport all travellers using only the edges in their best path. The model however can decide to use different edges (still part of the prescribed path) through the use of the variables y_e . These y_e model the number of people that were assigned to use type edge e of type t , but instead will be assigned to the set of underlying edges F of type $t - 1$. Note that in this way, these travellers could then again be reassigned from $f \in F$ to edges of type $t - 2$ etc. The structure of the constraints for the type edges depends on the type of the edge. For an edge e of type $t = 1$ in (11), there are no possibilities for rerouting passengers through y_e since, clearly, there are no edges of lower type. A similar argument for edges of type T_{\max} in (13) makes it clear that we can only reassign passengers from these edges, not to them.

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

Observation 2.1

For an arbitrary commodity k and an arbitrary 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:

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

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

Next, Observation 2.2 considers the sum of the demand for all commodities that are allowed to use some track edge e . Note that for any feasible flow F , this is equal to the sum of all the flow across e , i.e. the total flow on the edge e and all $f \in E_{t>1}^T | e \in R(f)$.

Observation 2.2

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

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

Lemma 2.5

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 will prove this lemma by induction on T_{\max} . First, consider the case in which $T_{\max} = 1$. From Observation 2.2 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

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

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 $e \in E_{t^*+1}^T$ implies that y_e is nonnegative. Clearly, now

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

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. \square

Lemma 2.6

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 Lemma 2.1 and Lemma 2.2 it is clear that we should show that feasible flows can be constructed from (X, Y) for artificial commodities $k = \{v, w\}$ for edges $\{v, w\} \in E^T$, with demand $\tilde{H}(k)$. We will prove this theorem using induction on T_{\max} . First, note that for $T_{\max} = 1$ all constraints of (11) are of the form $x(e) \geq \tilde{H}(e)$ for all $e \in E^T$. Since there is only one type of edges, we can thus set all

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

Thus, every commodity corresponds to an edge in E^T , and $F_f^k = 0$ for all $f \neq k$. Obviously, all flow restrictions (5)-(9) are satisfied.

Next, assume 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 (13) for the edges $e \in E_{t^*+1}^T$ are

$$x(e) \geq \tilde{H}(e) - y(e) \quad \forall e \in E^T$$

The total flow across 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 remaining 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 remaining edges. To show see that this is possible, note that by restriction (12) we know that

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

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

The previous two lemmas imply the following theorem.

Theorem 2.2

The problems MCF and IP_{XY} are equivalent.

Lemma 2.7

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

Proof. We will show that setting all y_e to the rounded down $\lfloor \tilde{y}_e \rfloor$ will result in a feasible solution (X, Y) . First, consider this rounding scheme for type edges e of type T_{\max} . The integrality of $x(e)$ and $\tilde{H}(e)$ ensures that $x(e) \geq \tilde{H}(e) - \lfloor \tilde{y}_e \rfloor$. Since $y_e \leq \tilde{y}_e$, all other restrictions will also remain satisfied. Moreover, using the integrality of $x(e)$, $\tilde{H}(e)$ and $\sum y_f$ thus also preserves the feasibility of the constructed solution for the remaining edges of types $T_{\max} - 1$ through $t = 1$. \square

2.4 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 will describe an alternative model. This model will not use the rerouting variable y_e , but will instead guide the routing by imposing additional restrictions.

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

Example 2.6

Consider the network displayed in Figure 6. From the four tracks, let us consider the constraints required for track $e = \{a, b\}$. The set S_2^e is a subset of $\{f_1, f_2\}$. Overall, this results in the following set of constraints:

$$\begin{aligned}
 x(e) &\geq \tilde{H}(e) \\
 x(e) + x(f_1) &\geq \tilde{H}(e) + \tilde{H}(f_1) \\
 x(e) + x(f_1) + x(f_2) &\geq \tilde{H}(e) + \tilde{H}(f_1) + \tilde{H}(f_2) \\
 x(e) + x(f_2) &\geq \tilde{H}(e) + \tilde{H}(f_2) \\
 x(g) &\geq \tilde{H}(g) \\
 x(g) + x(f_1) &\geq \tilde{H}(g) + \tilde{H}(f_1) \\
 x(h) &\geq \tilde{H}(h) \\
 x(h) + x(f_2) &\geq \tilde{H}(h) + \tilde{H}(f_2)
 \end{aligned}$$

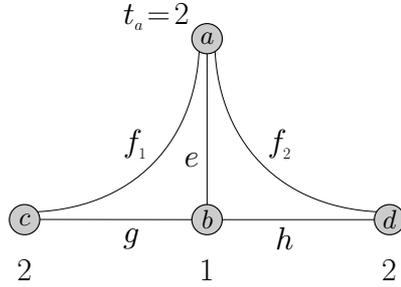


Figure 6: Consider the constraints needed for $e = \{a, b\}$

The integer programming formulation IP_X for ECP reads as follows

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

$$\text{s.t. } x(e) + \sum_{t>1} \sum_{f \in S_t^e} x(f) \geq \tilde{H}(e) + \sum_{t>1} \sum_{f \in S_t^e} \tilde{H}(f) \quad \forall e \in E_1^T, \forall S_2^e, \dots, \forall S_{T_{\max}}^e \quad (17)$$

$$x(e) \in \mathcal{C} \quad \forall e \in E^T \quad (18)$$

where the sets S_t^e for a given edge e of type 1 are such that $S_t^e \subseteq \bigcup_{f \in S_{t-1}^e} \{g \in E_t^T \mid R(f) \subseteq R(g)\}$ 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 (17) enforces sufficient capacities on edge e together with the type edges in the sets S_t^e for $t = 2, \dots, T_{\max}$.

Note that a constraint is added for all possible sets $S_2^e, \dots, S_{T_{\max}}^e$. Finally, feasible values for $x(e)$ are enforced by the set $\mathcal{C} \subset \mathbb{N}$, which represents the valid capacities that can be provided on an edge.

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

Lemma 2.8

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 edge $e \in E_1^T$, and with the sets $S_1^e, \dots, S_{T_{\max}}^e$. Recall restriction (17) for edge e :

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

Now, consider an arbitrary collection $S_2^e, \dots, S_{T_{\max}}^e$ of subsets, i.e. one of the constraints of type (17). Thus, since all y_e are nonnegative, the following holds

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

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

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

This first 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. \square

The inverse, extending a solution X of IP_X with appropriately chosen values for Y gives a feasible solution to IP_{XY} .

Lemma 2.9

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 2.5, we will again provide a scheme for constructing a suitable vector Y . Consider an arbitrary solution X of IP_X . Let us chose the y_e for any edge $e \in E_t^T$ as follows:

$$y_e = \left(\tilde{H}(e) - x(e) + \sum_{f \in E_{t+1}^T: R(e) \subseteq R(f)} y_f \right)^+ \quad (19)$$

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

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

Thus, substituting (19) for all y_f , we have to prove the validity of

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

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 (17) 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

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

As in the proof of Lemma 2.5, replacing the nested summations by the separate summations in the second equation can be done using the results from Lemma 2.4. Since this construction was done for an arbitrary solution X , this completes the proof. \square

The previous two lemmas imply the following theorem.

Theorem 2.3

The problems IP_X and IP_{XY} are equivalent.

2.5 Modelling LPP

As mentioned previously, LPP is a generalization of ECP. In ECP we have to assign capacities to individual edges, while in LPP capacities are assigned to lines, i.e. specially structured *sets* of edges.

We will use the same basic formulation of LPP as used by [CDZ98, Bus98, GHK01]. Moreover, an instance of LPP is described by a given network, as well as a set of potential lines

L . Every line l of L corresponds to a route (a simple path) through the track graph G , and the stations along its route at which it halts. This halting pattern of a line l 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 $e \in E^T$ for which $e \in E_{t_l}^T$ and $R(e) \subseteq l$. For every line we have to decide whether to deploy it and, if so, at what hourly frequency, and with how many carriages. 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 hourly capacity of a line l of type t is obtained by taking the product of the hourly frequency $f \in F_t$, the number of carriages $c \in C_t$ and the capacity of one carriage of type t , $\lambda_t \in \mathbb{N} : \lambda_t \cdot f \cdot c$. For formulating the line planning problem as an integer linear programming problem, we introduce a binary variable x for every $(l, f, c) \in N$, with the set of triples as $N := \{(l, f, c) | l \in L, f \in F_{t_l}, c \in C_{t_l}\}$ [GHK01]. Every $i \in N$ is associated with a combination $(l_i, f_i, c_i) \in N$. Given a solution, the total capacity available on a type track e is given by the sum of the capacities of all the passing lines, i.e.

$$x(e) \leftarrow \sum_{i \in N | l_i \text{ uses } e} \lambda_{t_i} f_i c_i x_i \quad \forall e \in E^T$$

	MCF	IP _{XY}	IP _X
# Vars.	$ E + \mathcal{O}(E^T ^2)$	$ E + E^T \setminus E_1^T $	$ E $
# Cons.	$ E^T + \mathcal{O}(E^T V)$	$ E^T $	$\mathcal{O}(2^{ E^T })$

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

The question remains, which of these models for ECP and thus for LPP to use for a given instance. Table 1 shows worst case figures for the number of variables and constraints for all models of ECP. To test the different models, we have done some computational experiments using three real-life instances. The results of this are discussed in the next section.

3 Computational results

We have used three real life instances to compare the computational complexity of solving them using the IP_X and IP_{XY} formulations. 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. The three

Instance:	$ T $	$ L $	$ V $	$ E $	$ E^T $
NS3600	3	64	28	27	50
NSNH	3	81	36	37	58
NSRandstad	3	331	122	138	204

Table 2: Instance characteristics.

networks are visualized in the graphs in Figure 7. The first two instances are rather small with respect to the number of nodes in the network. We have chosen these instances to compare the

practical use of two proposed models. At first glance, it seems that there will be a computation tradeoff between additional subset constraints in IP_X and the model structure of IP_{XY} . The track graph of NS3600 is a path. This is of great influence on the number of edges in E^T . From Lemma 2.3, it is easy to see that $|E^T| \leq |T| * |E|$, but even more importantly, the number of possible subsets for IP_X is also at most $|T| * |E|$, i.e. at most $|T|$ per edge $e \in E$. The other two instances introduce stations with a degree higher than 2 in the track graph, to test how the number of subset restrictions behaves.

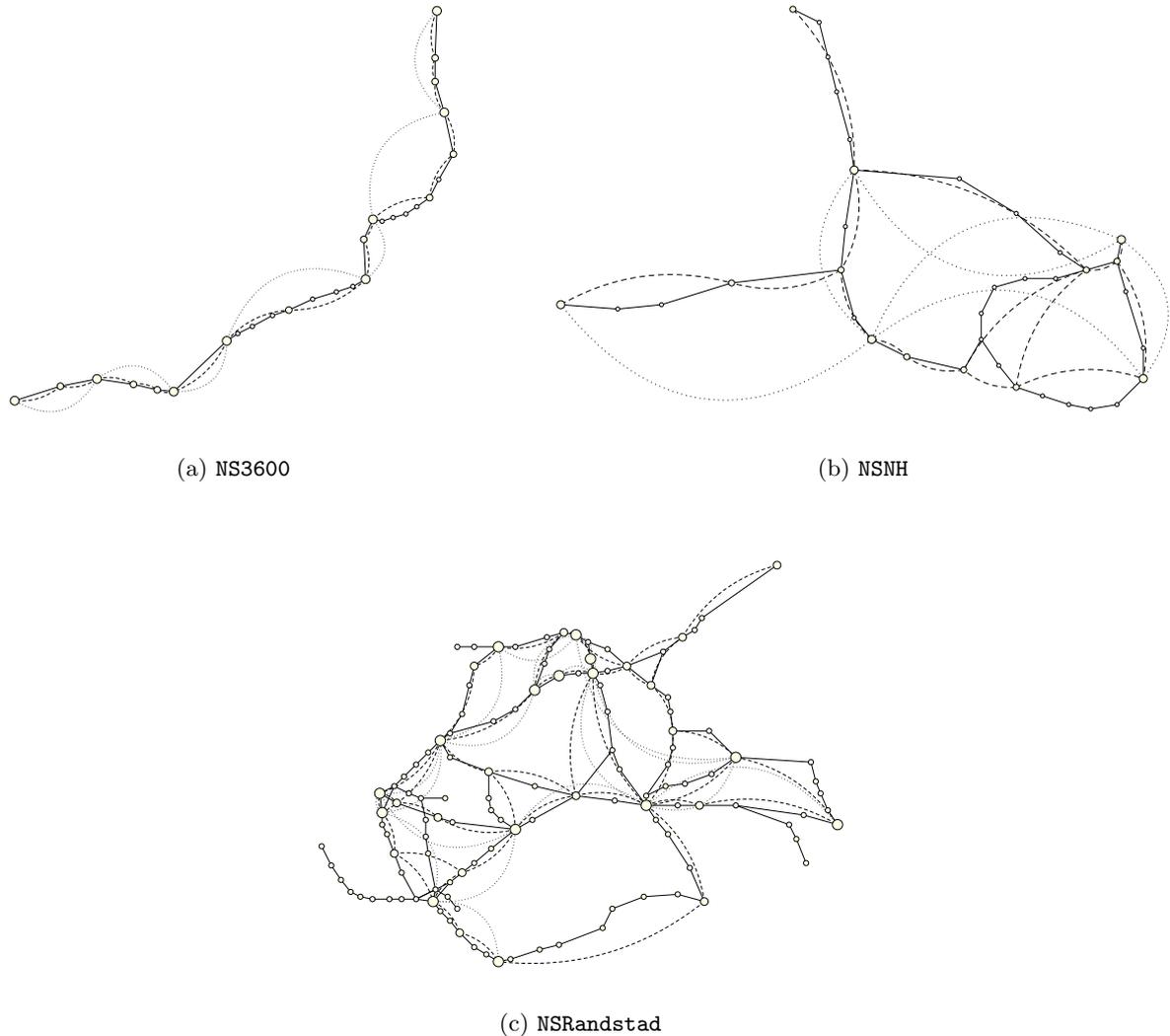


Figure 7: The type graphs for the instances NS3600 (7(a)), NSNH (7(b)) and NSRandstad (7(c)).

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

The results in Table 4 show that none of the IP_{XY} instances could be solved to optimality

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

Table 3: Statistics for the different instances and models.

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 conclusions can also be drawn from the other results.

Instance	Model	obj.	root LP	best LP	Gap	# sec.	# nodes
NS3600	IP_X	7520	7213	7520	0%	0.81	60
NS3600	IP_{XY}^\dagger	7520	6430	7173	4.61%	-	1206968
NSNH	IP_X	13760	13133	13760	0%	6.66	407
NSNH	IP_{XY}^\dagger	13760	12501	13313	3.25%	-	747874
NSRandstad	IP_X^\dagger	52480	48880	50510	3.75%	-	25008
NSRandstad	IP_{XY}^\dagger	55360	46146	48733	11.97%	-	104234

Table 4: Computational results. The dagger (\dagger) indicates that the time limit of 3600 seconds was reached.

4 Summary and conclusions

In this paper we have described several different (mixed) integer programming formulations for modelling the multi-type line planning problem. Where previous work, e.g. [Bus98, CDZ98] focuses on modelling LPP for exactly one type of trains and stations, we present generalizations of these models within a cost-minimizing setting. First, the general multi commodity flow formulation is introduced in Section 2.2. This formulation is then used to prove the validity of the two main formulations IP_X (Section 2.4) and IP_{XY} (Section 2.3). Using three real-life instances we compare the computational results for both formulations. From these test, we can conclude that the IP_X formulation, although with possibly an exponential number of constraints, outperforms IP_{XY} in all of the chosen instances.

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

References

- [Bus98] Michael Bussieck. *Optimal lines in public rail transportation*. PhD thesis, University of Braunschweig, 1998.
- [CDZ98] M.T. Claessens, N.M. van Dijk, and P.J. Zwaneveld. Cost optimal allocation of rail passenger lines. *European Journal of Operational Research*, 110:474–489, 1998.
- [CFT⁺97] A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P.L. Guida. Algorithms for railway crew management. *Mathematical Programming*, 79:125–141, 1997.
- [DMG98] F. Di Miele and G. Gallo. On the dispatching problem. Technical report, Dip. di Informatica, Univ. di Pisa, 1998.
- [GHK01] J.H.M. Goossens, C.P.M. van Hoesel, and L.G. Kroon. A branch-and-cut approach for solving line planning problems. METEOR Research Memorandum RM/01/016, University of Maastricht, 2001.
- [Nac99] K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. PhD thesis, Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig, Germany, 1999.
- [Odi97] M.A. Odijk. *Railway Timetable Generation*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 1997.
- [Olt94] C. Oltrogge. *Liniënplanung für mehrstufige Bedienungssysteme im öffentlichen Personenverkehr*. PhD thesis, Technical University of Braunschweig, 1994. In Dutch.
- [Sch93] A. Schrijver. Minimum circulation of railway stock. *CWI Quarterly*, 3:205–217, 1993.
- [SS94] A. Schrijver and A. Steenbeek. Dienstregelontwikkeling voor Railned (Timetable construction for Railned). Technical report, CWI, Center for Mathematics and Computer Science, Amsterdam, The Netherlands, 1994. In Dutch.
- [Zwa97] P.J. Zwaneveld. *Railway Planning, Routing of Trains and Allocation of Passenger Lines*. PhD thesis, Erasmus Universiteit Rotterdam, 1997.