



Operations Research

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Scheduled Service Network Design for Freight Rail Transportation

Endong Zhu, Teodor Gabriel Crainic, Michel Gendreau

To cite this article:

Endong Zhu, Teodor Gabriel Crainic, Michel Gendreau (2014) Scheduled Service Network Design for Freight Rail Transportation. *Operations Research* 62(2):383-400. <https://doi.org/10.1287/opre.2013.1254>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2014, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Scheduled Service Network Design for Freight Rail Transportation

Endong Zhu

App Annie, Beijing, China 100027, azhu@appannie.com

Teodor Gabriel Crainic

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation and Département de management et technologie, Université du Québec à Montréal, Montréal, Quebec H3C 3P8, Canada, teodorgabriel.crainic@cirreil.ca

Michel Gendreau

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation and Département de mathématiques et de génie industriel, École Polytechnique de Montréal, Montréal, Quebec H3C 3A7, Canada, michel.gendreau@cirreil.ca

This paper addresses the scheduled service network design problem for freight rail transportation. The proposed model integrates service selection and scheduling, car classification and blocking, train makeup, and routing of time-dependent customer shipments based on a cyclic three-layer space–time network representation of the associated operations and decisions and their relations and time dimensions. This paper also proposes a matheuristic solution methodology integrating slope scaling, a dynamic block-generation mechanism, long-term-memory-based perturbation strategies, and ellipsoidal search, a new intensification mechanism to thoroughly explore very large neighborhoods of elite solutions restricted using information from the history of the search. Experimental results show that the proposed solution method is efficient and robust, yielding high-quality solutions for realistically sized problem instances.

Subject classifications: scheduled service network design; rail freight transportation; capacitated multicommodity network design; slope scaling; ellipsoidal search.

Area of review: Transportation.

History: Received June 2011; revision received June 2013; accepted November 2013.

1. Introduction

Freight is moved in large part by *consolidation-based* carriers: railroads, less-than-truckload motor carriers, container ships, regular and express carrier services, etc. The fundamental idea of consolidation-based transportation is to group loads from different shippers, with possibly different origins and destinations, and to load them into the same vehicles for efficient long-haul transportation. The performance and profitability of such a system depend in large part on efficient and coordinated terminal and long-haul transport operations, as well as an offering of services meeting the cost and quality criteria of its potential customers. The *service network design problem* is at the core of the tactical planning process addressing these issues and producing the transportation plan of the carrier for the contemplated planning period (Ahuja et al. 2005, Cordeau et al. 1998, Crainic 2000, Crainic and Kim 2007).

Rail carriers generally implement a double consolidation policy: loaded and empty cars are grouped into so-called blocks, which are then grouped again to make up trains. Cars with different origins and destinations being present simultaneously in the same terminal are thus sorted and grouped into a *block*, which is moved as a *single unit* by a series of trains until its destination, where it is broken

down, the cars being either delivered to their final consignees or sorted for inclusion into new blocks. The associated operations and policies are denoted car *classification* and *blocking*, block *transfer* (from one train to another), and train *makeup*. Tactical planning for freight rail carriers then aims to select the train services to operate over the contemplated schedule length (e.g., the week), together with their frequencies or schedules (timetables), the blocks that will make up each train, the blocks to be built in each terminal, and the routing of the cars loaded with the customers' freight using these services and blocks (empty-car movements are also considered). The corresponding service network design problem aims to build this plan making the most efficient use of the railroad's assets to achieve its economic and customer-service performance objectives.

A rich literature exists on models and methods addressing these issues, as reviewed in the surveys indicated above. Most of them, however, either address a single or a limited number of issues, or make significant simplifying hypotheses. To the best knowledge of the authors, in fact, no model currently available in the literature addresses in an integrated, comprehensive formulation all of the tactical planning issues. Our goal is to answer this challenge and

present an *integrated scheduled service network design* methodology for rail freight transportation.

The major contribution of this paper is twofold. We propose what we believe to be the first comprehensive modeling framework for key tactical planning decisions, integrating service selection and scheduling, car classification and blocking, train makeup, and routing of time-dependent customer shipments. The framework is based on a three-layer space–time network representation of the associated operations and decisions, and their relations and time dimensions. Second, we propose an innovative solution methodology to address the large-size mixed-integer programming (MIP) formulation, which integrates exact and metaheuristic principles. The proposed matheuristic is based on *slope scaling*, long-term-memory-based perturbation strategies, and a new intensification mechanism, named *ellipsoidal search*, which thoroughly explores very large neighborhoods of elite solutions defined using information from the history of the search. The form of this restricted space inspired the name of the new procedure. Two variants of the matheuristic are presented: in the first, all potential blocks to be considered must be specified by the decision maker, but the second includes a dynamic block-generation mechanism, thus allowing one to consider only a limited number of blocks to start with, since useful blocks will be generated as needed. Experimental results show the matheuristic to be efficient and robust, as it yields high-quality solutions for realistically sized problem instances.

It must be emphasized that the interest of these contributions goes well beyond rail transportation. Our methodological framework encompasses the general field of service network design for consolidation-based transportation. It also opens interesting avenues for tackling multilayer design problems found in logistic (e.g., in city logistics), production, and telecommunication applications. The proposed solution approach could thus serve as a basis for developing effective solution methods for such problems.

This paper is organized as follows. Section 2 describes the problem setting and reviews the related literature. The three-layer space–time network representation for the scheduled service network design problem is introduced in §3, whereas the MIP formulation is presented in §4. The proposed solution method is introduced in §5, experimental results are analyzed in §6, and we conclude in §7.

2. Problem Setting and Literature Review

We initiate this section with a brief description of the freight rail transportation system and its main components, operations, and tactical planning challenges, and conclude by stating the particular problem addressed in this paper. The second part of the section is dedicated to a review of related literature.

Railroads are complex transportation systems where several major components interact and compete for resources. The infrastructure of the system is made up of a large number of terminals and rail tracks linking them. Most of these

terminals are stations where demand originates and terminates. A much smaller number are denoted *yards* and are specially equipped to handle large quantities of cars, sorting and grouping them for long-haul transportation, as well as to make up and disassemble trains. The term *classification* (or *marshalling*) *yard* is also often used to emphasize the major car-handling role of these facilities.

Customer demand takes the form of a number of cars of a type appropriate to the commodity that needs to be moved, to be shipped from an origin station to a destination one. The appropriate number of empty cars is delivered for loading to the customer (assuming it is connected to the rail network, or to a designated station, otherwise). Empty cars are generally delivered from a designated yard, and the loaded cars are moved back to the same yard or to a different one as appropriate for the long-haul movement toward the destination that follows. Most origin and destination stations are on secondary lines. Even when this is not the case, and the station is located on a main trunk line, the movement of loaded and empty cars between stations and their respective designated yards is generally performed by local trains (also called feeder trains) whose scheduling is usually not within the scope of the tactical planning process designing the long-haul service network. We follow this practice in this paper and assume that all demands are specified at the appropriate origin and destination yards.

Movements of freight on the rail network are performed by train services. A *train* is composed of one or more engines providing power and a series of cars. Each train has a particular origin yard where it is made up and a destination yard where it completes its journey, delivers all the cars currently hauled, and liberates the engines. A number of intermediary stops may be included in its route, the train delivering and picking up blocks at each one.

When the volume shipped between two stations in the systems is significant (i.e., the equivalent of at least a full train) and regular, railroads run dedicated trains nonstop from the origin to the destination of the freight. To take advantage of economies of scale, however, most shipments are handled according to consolidation principles, trains hauling groups of cars of different customers, with different origins and destinations. Then, schematically, cars at their origin yard are sorted, *classified* is the term generally used, and grouped with other cars with potentially different origins and destinations into particular blocks. The *block* is then handled as a single unit from its origin yard, where it is formed, to its destination yard, where it is taken apart, its cars being then either delivered to the respective consignees (the cars are at their final destination) or reclassified and blocked for the next part of their trip. Trains are thus made up of blocks and, when appropriate, it is blocks that are picked up and delivered at intermediate stops. Blocks may thus be *transferred* (*switched*) from one train to another. This double consolidation organization reduces car handling activities at yards.

Operations are constrained by the physical characteristics of the infrastructure and the operational policies of the railroad. Thus, for a given time period, classification-yard operations are limited by the yard capacity in terms of numbers of cars that may be classified, blocks that may be built, trains that may be made up, and so on. Similarly, the capacity of the rail tracks limits operations with respect to the number of trains that may operate “simultaneously” on a given track segment, as well as to the total weight (length also, sometimes) a train may haul on the track segment. The length and weight of trains are thus limited, and lower and upper limits may be imposed on blocks.

Service network design addresses the integrated planning of these operations, and aims for an allocation of resources to activities that fulfills the economic and customer-service objectives of the railroad. Several decisions are intertwined in this process, which yields an operating (the terms “load” and “transportation” are also used) plan and schedule:

- *Service selection and scheduling.* A *service* corresponds to a train route and type that may be operated, where the route specifies the origin, destination, sequence of yards visited, and physical route among them, whereas the type includes the planned speed, capacity, and priority of the train. Scheduling is performed for a given *schedule length* (e.g., one week) to be repeated for a certain *planning horizon* (e.g., a season). Schedules may be indicative, e.g., the frequency is only specified assuming a more or less uniform distribution of departures over the schedule length. Alternatively, a timetable may be specified for each service indicating arrival and departure times for each yard on the service route.

- *Blocking (and classification).* Blocking specifies the car classification policies at each yard and the corresponding blocks to be built.

- *Train make up.* Train make up determines the blocks the service hauls out of its origin yard and, eventually, the blocks it drops and picks up at intermediary stops.

- *Freight routing.* Freight routing specifies for each particular demand the itinerary used to transport its cars from origin to destination: the sequence of blocks and train services, and the corresponding yard operations. In this context, *demand* refers both to forecast customer requirements and to an estimation of the associated needs for empty car repositioning.

Service network design is thus a complex problem in most cases. It involves many selection (services, blocks, schedules) and routing (demand loads) decisions, each with network-wide impacts and all strongly and complexly linked in their economic, time, and space dimensions. An integrated model is required to address this problem in a comprehensive way. Most contributions in the literature, however, address a limited number of issues only.

Early contributions include the train routing and makeup model of Assad (1980) and the train routing and scheduling model of Morlok and Peterson (1970). Service selection is often addressed in two steps, service routes

and frequencies are first determined (e.g., Marín and Salmerón 1996, Goossens et al. 2004), train timetables being then constructed based on the resulting routing pattern (e.g., Brännlund et al. 1998; Caprara et al. 2002, 2006). Newman and Yano (2000) proposed a train scheduling model, which also assigned containers to trains, for the rail portion of intermodal shipments. Huntley et al. (1995) developed a computerized routing and scheduling system for CSX Transportation, whereas Ireland et al. (2004) developed a planning system for Canadian Pacific Railway that brought together several separate procedures without building a comprehensive model.

Blocking has often been addressed as a separate problem, assuming that train services were already selected. Bodin et al. (1980) proposed one of the first such models, a nonlinear MIP formulation. Newton et al. (1998) and Barnhart et al. (2000) formulated the blocking problem as a network design problem, with nodes and arcs representing classification yards and candidate blocks, respectively. No fixed costs were associated with blocks, but several capacity restrictions were considered to limit the number of blocks and the total volume of freight processed at each yard. A path formulation and a branch-and-price solution approach (Barnhart et al. 1998) was proposed in the former paper, whereas a dual-based Lagrangian relaxation was used in the latter to decompose the problem into easier-to-address subproblems: a continuous multicommodity flow problem and an integer block formulation that selected blocks satisfying yard capacity constraints (addressed by branch and cut). Ahuja et al. (2007) proposed a large neighborhood search algorithm for their mixed-integer blocking formulation aimed at addressing large problem instances.

Crainic et al. (1984) presented what was probably the first service network design model addressing simultaneously the selection of services and their frequencies, car classification and blocking, train makeup, and freight routing. Crainic and Rousseau (1986) generalized the model for the tactical planning of consolidation-based multicommodity, multimode freight transportation systems. The model took the form of a path-based, nonlinear network design formulation accounting for congestion phenomena in yards and on rail tracks, as well as for trade-offs between operating and time-related costs. A heuristic solution method was used to address realistically dimensioned problem instances derived from the case of a large North American railroad. It was a “static” formulation, however, assuming a uniform distribution of departures and indirectly selecting blocks through nonlinear accumulation delay functions.

Haghani (1989) attempted to combine train routing and scheduling, makeup, as well as empty car distribution on a space–time network with fixed travel times and prespecified traffic rules. A heuristic was used to address a somewhat simplified version of the model and illustrate the interest of integrated planning. The model proposed by Keaton (1989, 1992) aimed to determine which pairs of yards to connect by direct services, and whether to offer more than one train

a day, as well as the routing of freight and the blocking of rail cars. The service network was made up of one network for each pair of yards in the system with positive demand. Links represented trains and connections in yards, as well as a priori determined blocking alternatives. Blocking was not a direct decision, however, integer train connections and continuous car flows representing the decision variables. Solutions were obtained by using a combination of Lagrangian relaxation and various heuristics based on particular operation rules. Results were mixed. Although the model was used to perform a number of analyses on relatively small systems, convergence difficulties were also reported. Gorman (1998a) proposed a model aimed at the design of a scheduled operating plan that followed the particular operation rules of a given railroad. Model simplifications had to be introduced to achieve a comprehensive mathematical network design formulation, but the solution method was innovative. A hybrid metaheuristic, a tabu-enhanced genetic search, was used to generate candidate train schedules, which were evaluated on their economic, service, and operational performances. On relatively small but realistic problems, the metaheuristic performed well and was used for strategic scenario analysis for a major North American railroad (Gorman 1998b). All of the previous contributions modeled the blocking process through classification costs, with no explicit blocking decision variables.

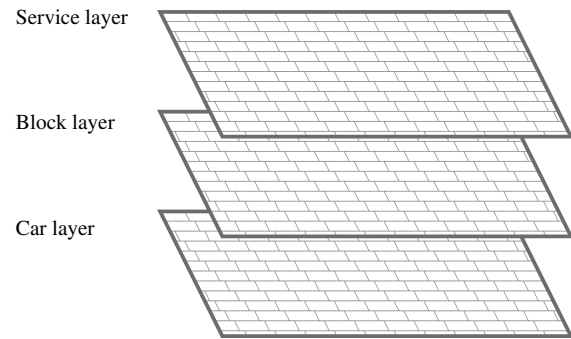
Andersen et al. (2009a, b), Pedersen and Crainic (2007), and Pedersen et al. (2009) added a new dimension to railroad tactical planning by introducing asset-management considerations into service network design models. The latter two contributions focused on the management of one asset type, engines, namely. A larger palette of issues was addressed in the first two papers, which also included scheduling aspects, cyclic schedules being constructed for services and assets. Classification and blocking issues were not addressed, however.

None of the previous contributions integrates all of the major tactical planning issues into a single scheduled service network design formulation. Our objective is to address this challenge. The next two sections detail our modeling framework proposal, which takes the form of a mixed-integer network design formulation on a cyclic space–time network that explicitly integrates decisions on service selection and scheduling, car classification and blocking, train makeup, and traffic routing.

3. Three-Layer Space–Time Network

This section is dedicated to the presentation of the three-layer space–time network structure that we propose to describe the operations and tactical decisions of a rail freight transportation system. This structure is built starting from the rail network on which operations are performed, represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertices $v \in \mathcal{V}$ representing yards, and directed links $e \in \mathcal{E}$ representing rail track sections.

Figure 1. Three-layer space–time network structure.



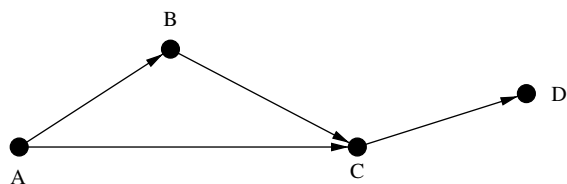
A number of measures associated with the vertices and links of \mathcal{G} represent limitations on the operations of the system. The volume of activities a classification yard may perform during any time period is limited by its physical characteristics, layout, number of tracks, engine or gravity-based sorting, etc., as well as available resources in terms of workforce, yard engines, and so on. To represent these limitations, we define the *classification capacity*, u_v^C , as the number of cars that may be classified at yard $v \in \mathcal{V}$, and the *blocking capacity*, u_v^B , as the number of blocks that may be built at the yard. Assuming, for simplicity of presentation, that a blocking track is dedicated to each block being built, the latter measure is given by the number of blocking tracks of the yard. Similarly, the track section *service capacity* u_e represents the maximum number of trains that may be present on link $e \in \mathcal{E}$.

Time-dependent issues in network design are generally addressed through space–time networks representing the occurrences of objects, events, and decisions at particular time moments. We follow this approach in this paper, and propose a three-layer structure to represent adequately the double consolidation activities proper to railroad transportation, cars into blocks and blocks into trains, as well as the three types of rail flow movements, namely, train services, blocks, and cars. The structure is illustrated in Figure 1.

The time dimension is the same in all layers. The schedule length is divided into \mathbf{T} equal-length time *periods* starting at time $t = 0, \dots, \mathbf{T} - 1$. The schedules are cyclic, that is, period 1 follows period \mathbf{T} . Yards are represented at each period through two nodes: an IN node where cars, blocks, and services arrive at the yard, and, symmetrically, an OUT node from where they leave the yard. Consequently, the total number of nodes in each layer is equal to $2 \times$ number of yards \times number of periods. Denote by \mathcal{N}^S , \mathcal{N}^B , and \mathcal{N}^C the sets of nodes in the service, block, and car layers, respectively.

The link set of the space–time network is the union of the link sets in each layer plus the vertical links connecting them. Yard operations in all layers are represented through various instantiations of *handling* links between IN and OUT nodes, and *holding* links between two consecutive IN or OUT nodes. The service layer also includes links

Figure 2. A simple rail network.



representing interyard train movements, defined between OUT and IN nodes of different yards at different periods. The attributes of nodes and links depend upon the layer they belong to and are specified later in the corresponding sections. The *temporal length* (or, simply, *length*) definition is, however, the same for all links: the number of time periods covered by the link. Vertical links have no duration. The up and down links between two consecutive layers represent the result of the yard car \leftrightarrow block and block \leftrightarrow train consolidation/deconsolidation activities. The three layers are described in the following subsections, which also introduce the definitions and notation used to formulate the proposed tactical planning model. Concepts are illustrated on a simple rail network with four yards and four directed track sections shown in Figure 2.

3.1. Service Layer

Links in the *service layer* represent train operations and include *moving* and *stop* links. A moving link $a \in \mathcal{A}^{\text{SM}}$ stands for a direct train movement from the OUT node of its origin yard at the departure time to the IN node of its destination yard at the arrival time (constant travel times are assumed for simplicity of presentation). The set $\mathcal{E}(a)$ includes the rail tracks defining the associated physical route. Figure 3 illustrates service-layer links defined on the rail network of Figure 2. Trains may move at different speeds on the same physical routes, which yields links of different lengths, e.g., the links (i_5, i_6) and (i_5, i_7) between yards B and C in Figure 3. Parallel moving links, e.g., with the same length between the same origin and destination nodes, represent train movements following different physical routes between the two yards with the same departure and transit times. Links a_1 and a_2 from node i_1 to node i_2 illustrate this case, where the former follows the route $A \rightarrow B \rightarrow C$, whereas the latter goes directly from A to C. A stop link $a \in \mathcal{A}^{\text{SR}}$ connects an IN node to the OUT node of the same yard at the next period, representing the time spent by a train at an intermediate yard to drop and pick up blocks and prepare for departure, e.g., links (i_4, i_5) and (i_7, i_8) in Figure 3.

Let \mathcal{S} be the set of possible services among which the selection is to be performed. A *service* $s \in \mathcal{S}$ is then defined as a path from $o(s) \in \mathcal{N}^{\text{S}}$, the OUT node of its origin yard, to $d(s) \in \mathcal{N}^{\text{S}}$, the IN node of its destination yard, through a set of moving, $\mathcal{A}^{\text{SM}}(s) \subset \mathcal{A}^{\text{SM}}$, and stop links. Service s_1 of Figure 3, which leaves yard A with destination D, with stops at B and C, and is made up of three moving and two

stop links, $i_3 \rightarrow i_4 \rightarrow i_5 \rightarrow i_7 \rightarrow i_8 \rightarrow i_9$, illustrates the definition. Each service is also characterized by a departure period, a (fixed) travel time, and a “fixed” cost c_s^{F} representing the cost of supplying the service (power, crew, and so on), as well as making up and disassembling the train. The capacity of the service s , u_s , indicates the maximum number of cars that may be hauled by a train.

Let $\mathcal{L}(s)$ be the set of sections of service s , each *section* $l \in \mathcal{L}(s)$ defining the service that s provides between two consecutive, but not necessarily adjacent, yards on its route. There are thus 6 sections for service s_1 in Figure 3: $(i_3 \rightarrow i_4)$, $(i_5 \rightarrow i_7)$, $(i_8 \rightarrow i_9)$, $(i_3 \rightarrow i_4 \rightarrow i_5 \rightarrow i_7)$, $(i_5 \rightarrow i_7 \rightarrow i_8 \rightarrow i_9)$, and the service itself. For each section $l \in \mathcal{L}(s)$, $s \in \mathcal{S}$, let $o(l), d(l) \in \mathcal{N}^{\text{S}}$ be its origin and destination yards, respectively, $\mathcal{A}^{\text{SM}}(l) \subseteq \mathcal{A}^{\text{SM}}(s(l))$ its set of moving links, and $s(l)$ the indicator function giving the service it belongs to.

3.2. Block Layer

Let \mathcal{B} denote the set of potential blocks. A block $b \in \mathcal{B}$ is built at its origin yard $o(b) \in \mathcal{N}^{\text{B}}$ and is moved until its destination yard $d(b) \in \mathcal{N}^{\text{B}}$ by one or more services, service-to-service block transfers being performed at the intermediary yards on its route in the latter case. A fixed cost c_b^{F} is associated with building block b at its origin yard and performing the block transfers (if any). The length of the blocking track assigned to block b yields its capacity u_b in terms of the number of cars it may carry.

Block yard operations are represented through *transfer* and *transfer-delay* links in the *block layer*, collected in sets \mathcal{A}^{BT} and \mathcal{A}^{BH} , respectively. A transfer link is a handling link representing the actual transfer procedure, i.e., taking the block off a service and attaching it to another one, and thus connects the IN and OUT nodes of the transfer yard with a length of one period. Transfer-delay links are holding links representing the time the blocking track is assigned at the origin yard—given as $h(b)$ periods (transfer-delay links)—as well as the time required to connect to the next service, to be determined by the model and algorithm. Arcs (i_7, i_8) and (i_6, i_7) in Figure 4 illustrate a transfer and a one-period transfer-delay link, respectively.

Block movements occur on sections in the service layer, the set $\mathcal{L}(b) \subset \mathcal{L}$ containing the sections used by block b . Figure 4 displays the projection of some of these links, e.g., set $\{(i_3, i_6), (i_8, i_9)\}$ representing the two sections of a block formed at yard A and departing from it at period 1 on one service, transferred and delayed one period at yard C, and arriving at destination yard D on a different service at time 8. The yard and long-haul block operations are connected by interblock links, illustrated by the vertical links of Figure 5, which represent attaching/detaching blocks to/from services. An upward *attach-block* link connects an OUT node in the block layer to the corresponding OUT node in the service layer and represents the result of adding the block to the service at the given period. Symmetrically, a downward *detach-block* link connects IN nodes of the

Figure 3. Service layer.

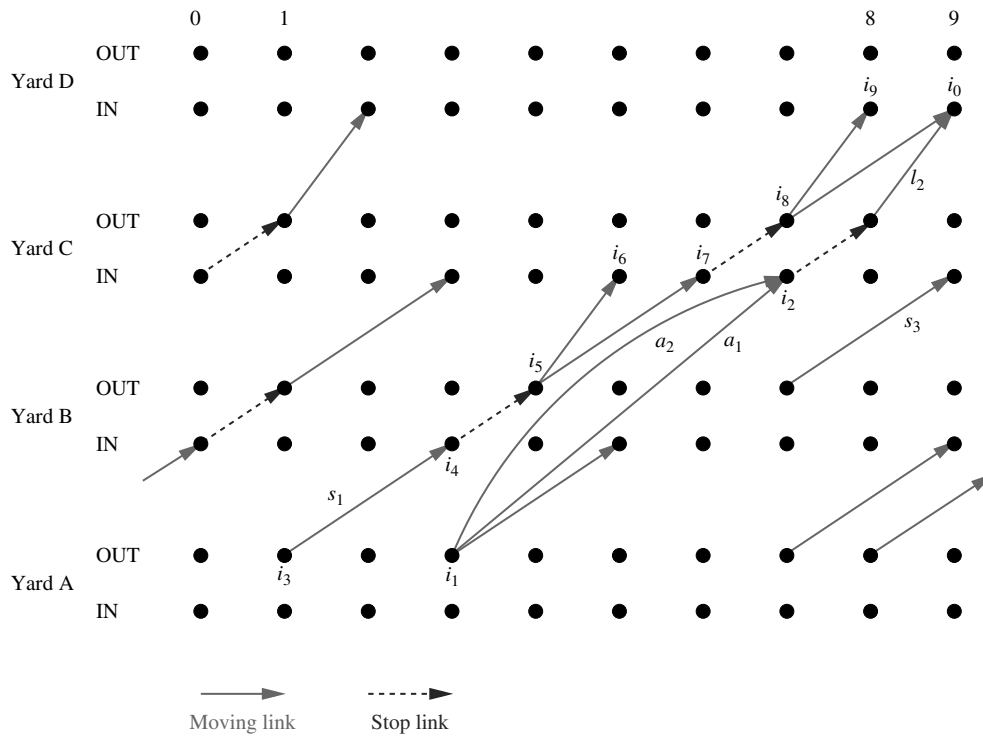
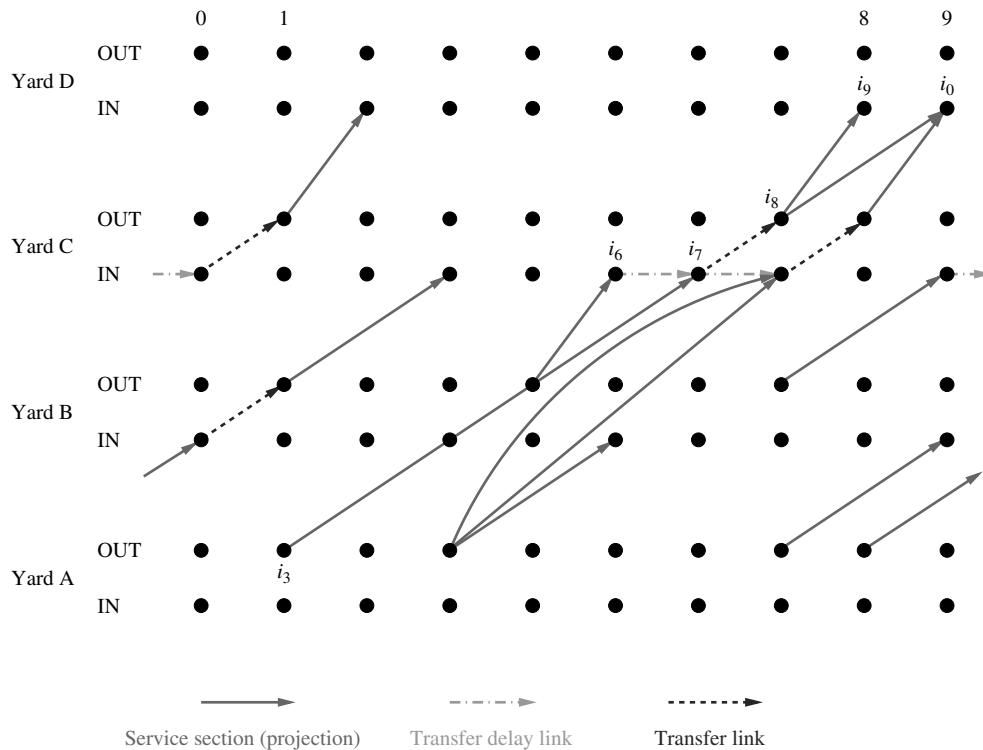


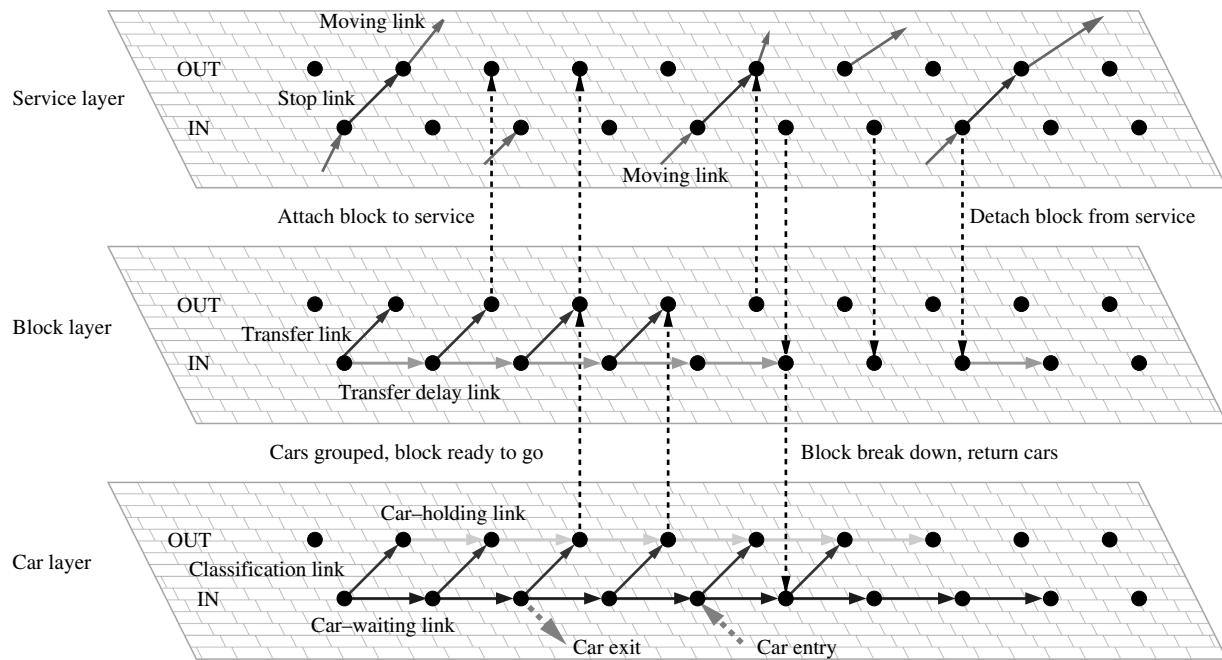
Figure 4. Block layer.



service and block layers, standing for taking the block off its current service and making it available either for transfer or dismantling (if at destination). The route of a block is thus defined as a path in the block and service layers

made up of $h(b)$ transfer-delay links at the origin yard, followed by a sequence of attach-block, service section, detach-block, transfer, and transfer-delay links describing its journey and the associated handling operations.

Figure 5. Main link types of the three-layer space–time network.



3.3. Car Layer and Itineraries

Similar concepts are defined for the *car layer*, illustrated in Figure 6, the links in this layer standing for the operations performed on cars in yards. There are two holding-type link sets, representing waiting in the receiving tracks to be classified and in the classification tracks for a sufficient number of cars to accumulate to build the block, respectively, and a handling one for the car classification operation. The three corresponding *car waiting*, *holding*, and *classification* link sets are denoted \mathcal{A}^{CW} , \mathcal{A}^{CH} , and \mathcal{A}^{CC} , respectively. A waiting link joins two successive IN nodes of a yard, a classification link connects an IN node to the OUT node of the same yard at the next period, and a holding link connects two successive OUT nodes of a yard, as illustrated by links (i_1, i_2) , (i_2, i_4) , and (i_4, i_5) , respectively. All of these links have a length equal to one time period. The classification capacity u_v^C of a yard $v \in \mathcal{V}$ is associated in this representation to its classification links, i.e., $u_a = u_v^C$ for $a \in \mathcal{A}^{CC}$. The interplay between this limit and the flow of cars requiring classification at the yard determines how long (i.e., the number of waiting links) cars will wait there.

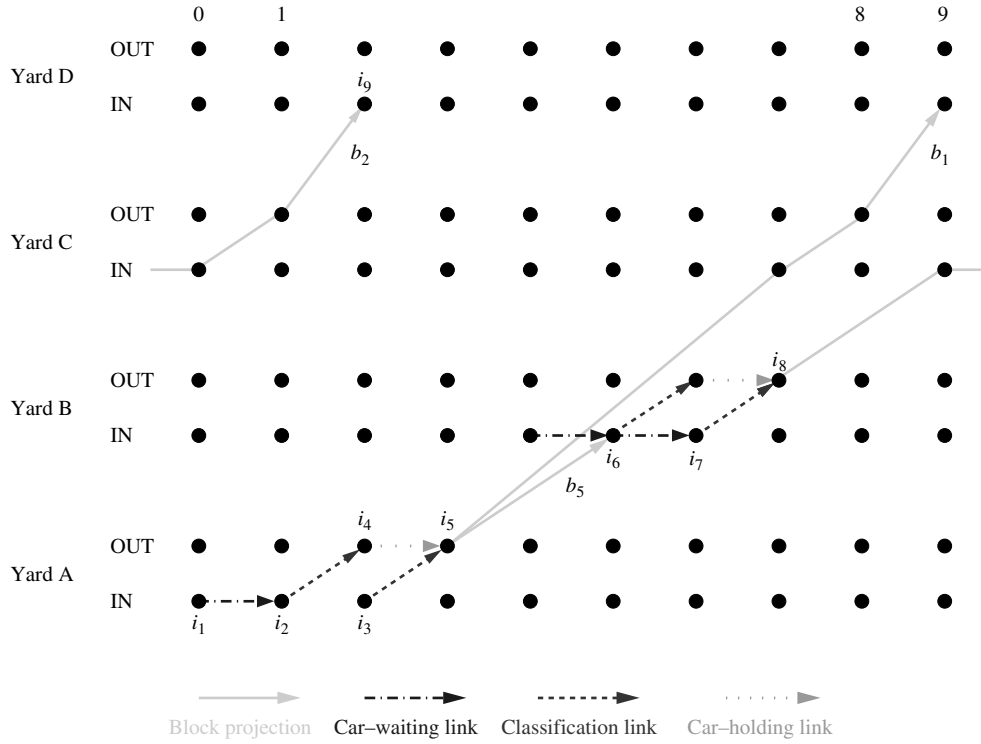
Cars are moved in blocks in the block layer, Figure 6 illustrating the projection of some of these movement links. “Vertical” links thus connect the two layers to represent the result of the car \leftrightarrow block activities at yards. Upward *group* links connect the OUT nodes in the car layer to their corresponding OUT nodes in the block layer and stand for the block formation activity. Symmetrically, a downward *breakdown* link is built from each IN node in the block layer to the corresponding IN node in the car layer to represent the activities related to blocks arriving at their

destinations to be broken down into cars. These links are illustrated in Figure 5, together with those in and out of the IN nodes of the car layer representing the car reception from and delivery to customers.

An *itinerary*, i.e., the journey of a group of cars of a particular demand, is then a path in the three-layer space–time network between two IN nodes in the car layer representing their origin and destination yards. Once at the IN node of their origin yard, cars wait for classification in the waiting links, are classified through a classification link, and reach an OUT node, where they wait for accumulation on car holding links. Once the block is formed, the traffic goes up to the block layer, where the block’s journey to its destination proceeds as described earlier on. At the destination, the block is unloaded and broken down, and the cars return to the car layer at the IN node of the associated yard and period. The journey ends if this node is the final yard of the demand. Otherwise, the cars go through classification and the block and service layers again until reaching their final destination.

The three-layer space–time network structure addresses the complete set of tactical planning activities and decisions. In particular, yard congestion phenomena, associated with higher workloads than the available capacity in a given period, are captured through the additional time spent by cars on car-waiting links (for yard classification congestion) and by blocks on transfer-delay links (for train congestion on line tracks). Selecting the best itinerary for each demand and the associated services, blocks, and schedule is the objective of the model presented in the next section.

Figure 6. Car layer.



4. Formulation

Let demand be specified by a set of *traffic classes* $\mathcal{P} = \{p\}$. Each traffic class $p \in \mathcal{P}$ is defined by an origin yard $o(p) \in \mathcal{V}$, a destination yard $d(p) \in \mathcal{V}$, a type of commodity $m(p)$, a quantity (number of cars) $w(p)$, a receiving time $r(p)$ at which cars become available at $o(p)$, and a due date by which cars must be received at $d(p)$. To simplify the notation on a cyclic network, we define the maximum *delivery time* $h(p)$ as the difference between the due date and the receiving time. We assume, as is usual in service network design settings, that \mathcal{P} includes empty car repositioning requests.

Let \mathcal{A} be the union of the links in all layers, and let c_{ap} represent the unit car cost for traffic class $p \in \mathcal{P}$ on link $a \in \mathcal{A}$. Notice that c_{ap} represents a hauling and time cost on interyard moving links, \mathcal{A}^{SM} ; a handling and time cost on car classification, \mathcal{A}^{CC} , and block transfer links, \mathcal{A}^{BT} ; and a time cost on the service stop, \mathcal{A}^{SR} , block transfer delay, \mathcal{A}^{BH} , car waiting, \mathcal{A}^{CW} , and car holding, \mathcal{A}^{CH} , links.

We define three groups of decision variables:

- service selection; $z_s = 1$ if service s is selected in the final design, 0 otherwise
- block selection; $y_b = 1$ if block b is built, 0 otherwise
- car flow distribution, x_{ap} , representing the number of cars of traffic class p on link $a \in \mathcal{A}$, with x_{bp} standing for the number of cars of traffic class p being part of block b

Define the indicator function as $\delta_{al}^{sb} = 1$ if the moving arc $a \in \mathcal{A}^{\text{SM}}$ of service s belongs to section $l \in \mathcal{L}(b)$ of block b , and $\delta_{al}^{sb} = 0$ otherwise. Define also $x_{asp} =$

$\sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}(b)} \delta_{al}^{sb} x_{bp}$, the total flow on moving link $a \in \mathcal{A}^{\text{SM}}$ of service s , and $x_{sp} = \sum_{a \in \mathcal{A}^{\text{SM}}} x_{asp}$, the total workload of service s with respect to the cars hauled on its moving links. The complete *integrated scheduled service network design (ISSND)* formulation, may then be written as follows:

$$\text{Minimize } \Phi = \sum_{s \in \mathcal{S}} c_s^{\text{F}} z_s + \sum_{b \in \mathcal{B}} c_b^{\text{F}} y_b + \sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{A}} c_{ap} x_{ap} \quad (1)$$

$$\text{s.t. } \sum_{a \in \mathcal{A}^+(n)} x_{ap} - \sum_{a \in \mathcal{A}^-(n)} x_{ap} = w_n^p \quad \forall n \in \mathcal{N}^{\text{C}}, \forall p \in \mathcal{P}; \quad (2)$$

$$\sum_{p \in \mathcal{P}} x_{bp} \leq y_b u_b \quad \forall b \in \mathcal{B}; \quad (3)$$

$$\sum_{p \in \mathcal{P}} x_{asp} \leq z_s u_s \quad \forall a \in \mathcal{A}^{\text{SM}}, s \in \mathcal{S}; \quad (4)$$

$$y_b \leq z_{s(l)} \quad \forall b \in \mathcal{B}, l \in \mathcal{L}(b); \quad (5)$$

$$\sum_{p \in \mathcal{P}} x_{ap} \leq u_a \quad \forall a \in \mathcal{A}^{\text{CC}}; \quad (6)$$

$$\sum_{b \in \mathcal{B}(v,t)} y_b \leq u_v \quad \forall v \in \mathcal{V}, \forall t \in \{1, \dots, \mathbf{T}\}; \quad (7)$$

$$\sum_{s \in \mathcal{S}(e,t)} z_s \leq u_e \quad \forall e \in \mathcal{E}, \forall t \in \{1, \dots, \mathbf{T}\}; \quad (8)$$

$$x_{ap} \geq 0 \quad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}; \quad (9)$$

$$y_b \in \{0, 1\} \quad \forall b \in \mathcal{B}; \quad (10)$$

$$z_s \in \{0, 1\} \quad \forall s \in \mathcal{S}. \quad (11)$$

The objective function (1) minimizes the total cost of the system for the schedule duration, that is, the total cost of selecting and operating services and blocks, waiting for operations, and moving the loaded and empty cars to satisfy demand.

Let $\mathcal{S}^+(n)$ and $\mathcal{S}^-(n)$ represent the sets of outward and inward links of node $n \in \mathcal{N}^C$. Constraints (2) then enforce the car flow conservation in the car layer and, thus, satisfaction of demand, where $w_n^p = w(p)$ if $n = o(p)$, $w_n^p = -w(p)$ if $n = d(p)$, and $w_n^p = 0$ otherwise.

Relations (3) and (4) are linking constraints enforcing the block and service moving-link car capacities, respectively, provided the corresponding blocks and services have been selected. Relations (5) are linking constraints as well, specifying that a block cannot exist unless all services providing the sections on its path have been selected.

Constraints (6) and (7) limit the yard car-classification and block-building workload at each period. Because building a block occupies a track for several ($h(b)$) periods, $\mathcal{B}(v, t)$ is the set of blocks being built simultaneously in yard v at period t . Similarly, constraints (8) enforce the limit on the number of trains that may simultaneously run on a link (track segment), $\mathcal{S}(e, t)$ standing for the set of such services for link $e \in \mathcal{E}$ in period t . Finally, constraints (9)–(11) specify the domain of each set of decision variables.

5. Solution Method

The model proposed above yields a mixed-integer formulation of large dimensions, both in the number of integer variables and the number of constraints. The problems belong to the general class of capacitated multicommodity network design (CMND) problems, which is known to be *NP*-hard (Magnanti and Wong 1984). Previous experience with CMND formulations showed that exact methods cannot efficiently address instances of meaningful dimensions and that metaheuristics integrating large-scale neighborhoods are required (Ghamlouche et al. 2003, 2004). They also showed, however, the interest of combining heuristics and methods taking advantage of the properties of the mathematical formulation of the problem (Hewitt et al. 2010).

We introduce such a metaheuristic for the ISSND formulation starting from the *slope-scaling* (SS) idea of Kim and Pardalos (1999), strengthened by Crainic et al. (2004) and Kim et al. (2006). Applied to the CMND, SS iteratively solves a linear approximation of the formulation, the resulting flow distribution being then used to enhance the current approximation of the fixed costs. One thus approaches the optimum solution of the CMND by solving successive linear approximation problems, the terms of the objective function being updated recursively. The perturbation of the linearization factors based on long-term search memories helps the procedure move out of local optima of rather mediocre quality.

The challenge in addressing the ISSND formulation is compounded by the two-layer structure of its design.

The problem and formulation display two sets of design decisions and variables, respectively, locked in a double-sided combinatorial relation, a train being made up of a series of blocks, while each block corresponds to a sequence of train sections. The main issues in defining a slope-scaling strategy for such a formulation are the selection of the set of design variables to linearize, the resolution of the resulting approximation problem, and the determination of feasible solutions to the ISSND formulation based on the solutions of the linearized approximation. The intuitive approach to the first issue is to apply the fundamental principle of SS and linearize both sets of design variables. We detail this basic SS strategy for the ISSND formulation in §5.1.

An alternative approach linearizes one set of design variables only. Although less intuitive, because the resulting approximation problem is still a mixed-integer CMND formulation, this approach takes advantage of the fact that the high design layer, trains on services, makes up the backbone of the tactical plan, whereas the second design layer, blocks, may be viewed and generated as sequences of train sections. It thus linearizes the service design variables only, and applies a metaheuristic generating blocks dynamically to address the resulting block-CMND problem. Section 5.2 details this dynamic SS procedure for the ISSND formulation.

Two procedures complete the proposed algorithm, a mechanism to perturb the linearization factors when the SS search stalls (§5.5), and a new intensification method, named *ellipsoidal search*, detailed in §5.4. The latter thoroughly explores the solution space, suitably restricted based on the characteristics of elite solutions and the long-term memories, around good solutions identified during the search and collected in a *reference set*.

At a glance, the overall procedure, displayed in Algorithm 1, starts by defining R-ISSND, a relaxed problem obtained by removing the capacity constraints on the design variables to be linearized. It then proceeds through search phases separated by perturbations of the linearization factors Λ . In each phase, a sequence of SS iterations finds a local best solution to the linearized formulation $AP(\Lambda)$, possibly generating new blocks; extracts a feasible solution to R-ISSND by selecting the design arcs with positive flows; and gradually updates the linearization factors. The sequence stops when no further improvement is obtained, i.e., when the flow (and block in the dynamic-SS case) patterns of two consecutive iterations are the same, indicating a local optimum has been reached, or no improvement is observed in the value of the R-ISSND feasible solution for $I_{\max}^{\text{nonimprove}}$ consecutive iterations (even though different solutions might have been obtained for the AP problem). When required, a feasibility-restoration procedure yields feasible ISSND solutions by solving a series of so-called *sliding problems* (§5.3). The ellipsoidal search is implemented before the perturbation of the linearization factors is triggered. The procedure repeats until it reaches

either the maximum computing time T_{\max} or the maximum number of SS iterations I_{\max} .

Algorithm 1 (Proposed matheuristic)

Set initial values for linearization factors Λ ;

Reference set = \emptyset ;

Define R-ISSND;

while Stopping criteria not met **do**

repeat

 Generate an approximation problem AP(Λ);

 Address AP(Λ):

For Basic_SS: Multicommodity capacitated network flow problem;

For Dynamic_SS: Block-CMND by tabu search, dynamically generating new blocks;

 Extract a feasible solution to R-ISSND;

 Update Λ ;

until No further improvement is obtained

 Generate a series of feasible solution to ISSND by solving sliding problems (updating the reference set);

 Intensify the search based on the reference set through the ellipsoidal search;

 Perturb linear factors Λ based on long-term memories.

end while

5.1. Basic Slope Scaling

Define the relaxed problem, R-ISSND, by removing the block-building (7) and service-on-link (8) constraints of the ISSND formulation involving design variables only. Let β_s , β_{as} , $\alpha_b \in \Lambda$ be the linearization factors associated with each service s , moving link $a \in \mathcal{A}^{\text{SM}}$ of service s (where the fixed cost of s is distributed over its moving links, i.e., each moving link gets a $c_s^F/|\mathcal{A}^{\text{SM}}(s)|$ cost) and block b , respectively. The approximation problem may then be defined as

$$\text{AP}(\alpha, \beta): \text{Minimize} \left\{ \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}^{\text{SM}}} \beta_{as} x_{asp} + \sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} \alpha_b x_{bp} + \sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{A}} c_{ap} x_{ap} \right\} \quad (12)$$

subject to car flow conservation (2) and handling (6) constraints, as well as the capacity restrictions on the numbers of cars hauled by services on their moving links (13) and grouped into blocks (14):

$$\sum_{p \in \mathcal{P}} x_{asp} \leq u_s \quad \forall a \in \mathcal{A}^{\text{SM}}, s \in \mathcal{S}; \quad (13)$$

$$\sum_{p \in \mathcal{P}} x_{bp} \leq u_b \quad \forall b \in \mathcal{B}. \quad (14)$$

AP(α, β) is a multicommodity capacitated network flow problem, which can be solved by the Simplex method. Yet, because AP(α, β) has to be solved repeatedly, and given the usually large dimension of the instances addressed, this approach may require excessive computing time (and memory). We therefore propose a heuristic based on the shortest augmenting-path algorithm. Define the residual capacity of a traffic-class itinerary as the minimum of the residual

capacities of its classification links, blocks, and sections. Flows are then assigned to the lowest cost itinerary with positive residual capacity. Preliminary experiments (§6.2) showed that the heuristic generates very good solutions within much shorter computing times compared with a state-of-the-art solver.

Addressing AP(α, β) yields a flow pattern \tilde{x} , leading to a feasible design for R-ISSNF by selecting only blocks \tilde{y} and services \tilde{z} with positive car flows. For the approximation to adequately reflect the total cost, its linearized term $\sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}^{\text{SM}}} \beta_{as} x_{asp} + \sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} \alpha_b x_{bp}$ should equal the corresponding ISSND design cost $\sum_{b \in \mathcal{B}} |\tilde{y}_b=1| c_b^F + \sum_{s \in \mathcal{S}} |\tilde{z}_s=1| c_s^F$. We therefore update the linearization factors for the selected services, sections, and blocks according to (15) and (16):

$$\alpha_b = \frac{c_b^F}{\sum_{p \in \mathcal{P}} \tilde{x}_{bp}} \quad \text{if } \tilde{y}_b = 1; \quad (15)$$

$$\beta_{as} = \frac{c_s^F}{\sum_{p \in \mathcal{P}} \tilde{x}_{asp}} \quad \text{if } \tilde{z}_s = 1. \quad (16)$$

Initial values for the linearization factors are computed according to the fixed-cost/capacity ratio, often used to obtain initial solutions to network design problems (Kim and Pardalos 1999): $\alpha_b = c_b^F/u_b$, $\forall b \in \mathcal{B}$ and $\beta_{as} = c_s^F/|\mathcal{A}^{\text{SM}}(s)|u_s$, $\forall a \in \mathcal{A}^{\text{SM}}$, $s \in \mathcal{S}$.

Long-term memories are updated following each slope-scaling iteration (§5.5) and are used to guide the perturbation and ellipsoidal-search phases, as well as to adjust penalties for yard or track capacity violations. These memories track four service and block utilization measures continuously over all iterations; no reinitialization is performed following perturbation and ellipsoidal-search phases. The first is the *open frequency*, f_i , $i = \{s, b\}$, recording the number of iterations for which the service or block was selected, respectively. The other three measures are based on the flow carried by services and blocks: *average car flow*, \bar{x}_i ; *peak car flow*, \hat{x}_i ; and *flow variability ratio*, $\mu_i = \bar{x}_i/\hat{x}_i$ ($\mu_i = 0$ when $\hat{x}_i = 0$), for $i = \{s, b\}$. The latter reflects the consistency of the loads a service or block carried, a value close to 1 indicating that, historically, car flows were close to the respective capacity; the variability of the flow carried by the respective service or block increases significantly as the ratio approaches 0. Notice that the service car flow measures are computed using the total workload x_{sp} of the service.

5.2. Block-Dynamic Slope Scaling

Only service design variables are linearized in this version, defining R-ISSND by removing the service-on-link (8) constraints of the ISSND formulation and setting $\Lambda = \{\beta_s\}$. The approximation problem then becomes (the initial set \mathcal{B} is populated by building a block for each service section)

$$\text{AP}(\beta): \text{Minimize} \left\{ \sum_{b \in \mathcal{B}} c_b^F \cdot y_b + \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} x_{sp} \beta_s + \sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{A}} c_{ap} \cdot x_{ap} \right\} \quad (17)$$

subject to constraints on car-flow conservation (2), car-to-block linking/capacity (3), car-classification capacity (6), and block-building at yards (7), as well as the train load capacity (18):

$$\sum_{p \in \mathcal{P}} x_{asp} \leq u_s \quad \forall a \in \mathcal{A}^{SM}, s \in \mathcal{S}. \quad (18)$$

Define c'_{bp} , the surrogate flow cost for traffic class p and block b , as the sum of the unit flow costs $c_{ap} + \beta_s$ over the (moving) links a of the service sections making up the block b . The objective (17) then becomes

$$\text{AP}(\beta): \quad \text{Minimize} \left\{ \sum_{b \in \mathcal{B}} c_b^F \cdot y_b + \sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} c'_{bp} x_{bp} \right. \\ \left. + \sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{A}^{CW} \cup \mathcal{A}^{CC} \cup \mathcal{A}^{CH}} c_{ap} \cdot x_{ap} \right\}, \quad (19)$$

and $\text{AP}(\beta)$ is a *block network design CMND* formulation. Addressing $\text{AP}(\beta)$ yields a feasible flow distribution \tilde{x} and a block design \tilde{y} . A feasible solution to R-ISSND may then be obtained by selecting services s used by blocks selected in \tilde{y} , i.e., $\tilde{z}_s = 1$ when one of the service sections of s is used by b , and $\tilde{y}_b = 1, \tilde{z}_s = 0$, otherwise.

The linearization factors β_s are updated according to Equation (20) to satisfy condition (21) for the selected services. As for the long-term-memory updates, the procedure proceeds as for the basic version (with the exception of f_b , of course):

$$\beta_s = \frac{c_s^F}{\sum_{p \in \mathcal{P}} \tilde{x}_{sp}}, \quad (20)$$

$$\sum_{p \in \mathcal{P}} \tilde{x}_{sp} \beta_s = c_s^F \tilde{z}_s. \quad (21)$$

We propose a cycle-based (Ghamlouche et al. 2003) tabu search (TS) heuristic with a dynamic block-generation mechanism to address the *block-CMND* formulation. The TS heuristic is based on the structure proposed by Zhu et al. (2009) for the direct service network design problem (see also Zhu 2011), enhanced to increase performance within a slope-scaling environment and to include the dynamic block generation capability.

The main idea of the cycle-based neighborhood is to improve a given solution by “closing” and “opening” sets of design arcs by redirecting flow on promising (in terms of reducing both fixed and variable costs) cycles. In applying this idea to the $\text{AP}(\beta)$, we integrate the generation of promising blocks into the neighborhood evaluation, that is, into the computation of such cycles. We now describe the main steps of the TS heuristic, which are summarized in Algorithm 2.

Algorithm 2 (Tabu search heuristic with block generation)

Require: An $\text{AP}(\beta)$ solution \tilde{x} and \tilde{y} ;

Require: Current block set \mathcal{B} ;

while Stopping criteria not met **do**

 Set $\text{MinCycle} = 0$;

for Each open block b in \tilde{y} **do**

 Compute the flow value γ ;

 Generate blocks between eligible OUT–IN node pairs through shortest-path algorithm on the (β, γ) -block layer;

 Add new blocks to set \mathcal{B} ;

 Construct the γ -residual network;

 Identify the lowest-cost cycle for block b and update MinCycle

end for

 Deviate flow on the MinCycle , update the block design and the long-term memory, and assign a tabu tag to blocks that changed their status (opened or closed);

 Distribute the car flow on the new block design, and restore feasibility (when needed);

if Current solution is better than or close to the best overall $\text{AP}(\beta)$ solution **then**

 Intensify the search;

else

if Improvement on current solution is negligible **then**
 Diversify the search;

end if

end if

end while

Let \mathcal{B} be the set of blocks available at the current iteration. The search space of the TS heuristic is made up of block designs in \mathcal{B} , and corresponding car flows, feasible for the approximated $\text{AP}(\beta)$ problem. The neighborhood of a solution in this search space contains all solutions (block designs) that may be reached by closing a currently open block b , as well as closing and opening other blocks arrayed in a cycle including b . Clearly, one desires to identify the cycle that offers the best prospect to decrease the objective value of $\text{AP}(\beta)$. Then, given the flow $\gamma = \sum_{p \in \mathcal{P}} x_{bp} > 0$ carried by b in the current block set \mathcal{B} , any cycle that permits to close b must offer the possibility to redirect at least γ units of flow. Moreover, among all such cycles that may be defined for block b , we are interested in the one that yields the largest cost decrease. An extended Bellman–Ford algorithm (Zhu et al. 2009) applied on a suitably defined γ -residual network searches this very large neighborhood and identifies least-cost candidate cycles.

The γ -residual network is based on the car-layer projection made up of block links corresponding to blocks in \mathcal{B} , as well as car classification, waiting, and holding links. A block from an OUT node i to an IN node j yields possibly two block links:

- a *forward* arc $(i, j)^+$ if the residual capacity of the block, with respect to block and service capacities, is larger than γ ; the cost of the arc equals the transportation cost for γ cars on the block, plus the fixed cost of the block when the block is currently closed;

• a *backward* arc $(j, i)^-$ if a block is open and with a total flow larger or equal to γ ; the negative cost (savings) of the arc equals the transportation cost for γ cars on the block, plus the fixed cost of the block if this is to be emptied by the move (i.e., the current flow = γ).

Forward arcs are built for all car waiting and holding links (these are uncapacitated), whereas backward links are built when the flow is at least γ cars. Classification arcs with positive flows are grouped and forward/backward arcs are added between each IN–OUT node pair to permit adding/subtracting flows on the in-yard paths of the corresponding station. The construction of the γ -residual network is illustrated in Figure 7.

The shortest-path algorithm identifies the lowest-cost cycle for block b , accounting for the temporal bounds imposed by the car flow conservation constraint enforcing timely delivery of demand. The cycle with the lowest cost, hopefully negative, among those of all blocks with positive flow is selected, and car flows are shifted along this cycle yielding a new flow distribution \tilde{x} (an illustration of the move is given in the electronic companion, available as supplemental material at <http://dx.doi.org/10.1287/opre.2013.1254>). The block design \tilde{y} is modified accordingly, yielding a new AP(β) solution.

New blocks are generated and added to \mathcal{B} to increase the probability of identifying improving cycles. New blocks therefore have to be low-cost and able to carry at least γ units of flow. Because blocks are moved by trains on service sections, blocks are generated on a (β, γ) -block network with links corresponding to service sections with at least γ residual capacity, plus block-transfer and transfer-delay links. The unit costs of these links are computed as averages over the set of commodities and are given by Equations (22) and (23), with the block fixed costs linearized by γ , and (24), respectively:

$$c'_l = \sum_{a \in \mathcal{S}^{\text{SM}}(l)} \left(\frac{\sum_{p \in \mathcal{P}} c_{ap}}{|\mathcal{P}|} + \beta_{s(l)} \right) + \sum_{a \in \mathcal{S}^{\text{SR}}(l)} \frac{\sum_{p \in \mathcal{P}} c_{ap}}{|\mathcal{P}|} \quad \forall l \in \mathcal{L}; \quad (22)$$

$$c'_a = \frac{\sum_{p \in \mathcal{P}} c_{ap}}{|\mathcal{P}|} + \frac{c_a^F}{\gamma} \quad \forall a \in \mathcal{A}^{\text{BT}}; \quad (23)$$

$$c'_a = \frac{\sum_{p \in \mathcal{P}} c_{ap}}{|\mathcal{P}|} \quad \forall a \in \mathcal{A}^{\text{BH}}. \quad (24)$$

A block is represented in the (β, γ) -block network by a path consisting of service sections connected by transfer and transfer-delay links. The set of such paths connecting an *eligible* OUT–IN node pair (i.e., the OUT and IN nodes are from different yards at different time periods) makes up a *block kernel* (or *parallel block set*). We then generate the least-cost path in each block kernel and add the corresponding blocks to set \mathcal{B} .

The lowest-cost cycle over all currently open blocks, *MinCycle* in Algorithm 2, identifies the neighboring solutions to which the algorithm moves by deviating γ units of

aggregated flow along the cycle, updating the block design, and reoptimizing the flow distribution over the new design. The shortest augmenting path heuristic introduced in §5.1 is used in this case as well. (This is different from Zhu et al. (2009), where the minimum cost flow problem was solved to optimality.) Particular origin-destination demands that cannot be feasibly delivered are sent on artificial links connecting the corresponding origin and destination nodes, and a restoration phase is implemented to regain feasibility. Restoration follows the same cycle-based idea, where cycles are constructed for artificial links with positive flow on graphs with γ values equal to the corresponding flow volumes.

Blocks that changed their status (from open to closed or vice versa) receive a tabu tag forbidding the reversal of this status for a tabu tenure randomly selected in a predefined integer range. In addition to this classical utilization of tabu memories to prevent cycling, we use them to prevent the search from exploring unfeasible regions. Each time a minimum-cost cycle is calculated, the blocks violating the block-building constraints are tagged with a tabu tenure, and we calculate the next nontabu cycle. A long-term memory is also updated recording the frequency of blocks open in feasible solutions obtained by the tabu search.

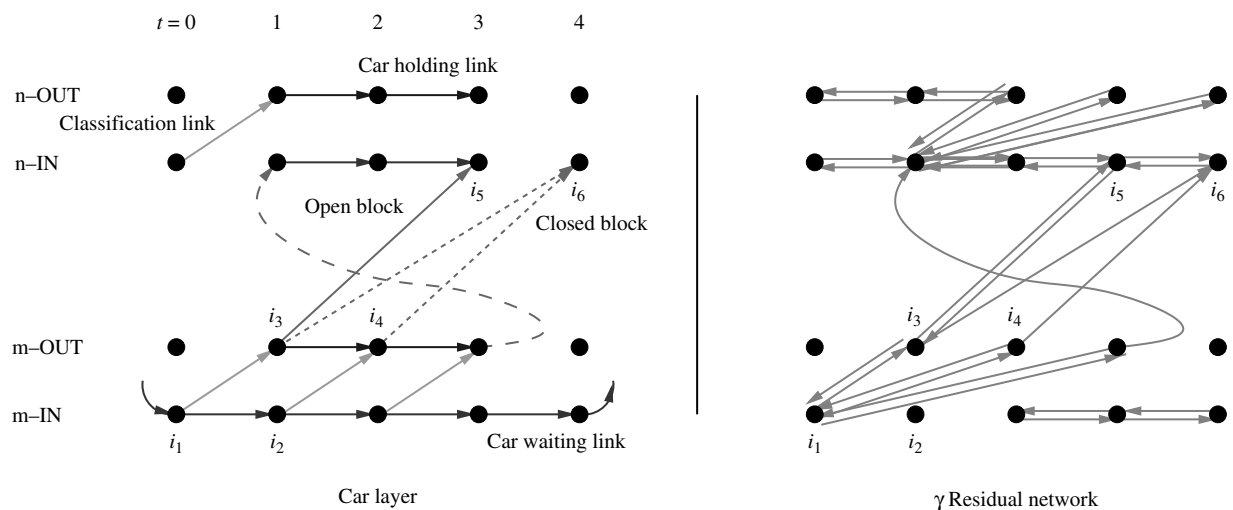
Two processes are included to direct the search into/out of specific domains. *Intensification* is triggered each time a new best AP(β) solution is found or when the value of the new current solution is close to the best AP(β) solution. Intensification proceeds similarly to the main tabu search but with a commodity-specific neighborhood, i.e., instead of working with aggregated flows, cycles of each commodity present on the block are evaluated with an assumption that the flows from other commodities are fixed. *Diversification* is called upon when there were no or only minor improvements in the last moves. To diversify the search, the long-term memory is exploited, and a small number (e.g., 15%) of frequently open blocks are forced to close. The flows are then redistributed on the new design, feasibility is restored, when required, as described previously, and the tabu search continues from this diverse solution.

To conclude this section, notice that the tabu search procedure with dynamic block generation offers a number of benefits. From a computational efficiency point of view, it facilitates a “continuous” resolution of the series of approximation problems, as the search at the current SS iteration may start from that of the previous iteration. Moreover, from an application point of view, the block generation procedure provides the means to consider application-specific restrictions, e.g., resource availability or characteristics of particular facilities, without the need to explicitly integrate them into the model formulation.

5.3. Restoring Feasibility

A feasible solution to the R-ISSND may not be feasible for the original ISSND with respect to the block-building (7) and service-on-link (8) capacity constraints for the basic

Figure 7. Construction of the γ -residual network.



SS version, or only with respect to the latter for the block-dynamic version. The repair procedure we propose is based on the idea of “sliding” a number of selected services and blocks forward or backward in time, to distribute the workload in excess among “copies” of those services and blocks, which are assumed to be “good” because they are part of the current local optimum.

Define the *parallel neighbors* of a service (block) as the services (blocks) with the same route and transit time, but departing at the previous and the subsequent period. The *sliding problem* corresponding to a given solution is then defined as the ISSND model restricted to the services (and blocks for the basic version) selected in the current solution, plus their respective parallel neighbors.

The sliding-restricted problem has a relatively small number of integer variables and may be addressed efficiently by an MIP solver. We therefore continue performing the procedure once a feasible solution is obtained to further improve it. When the sliding procedure produces no feasible solution, we penalize the services and blocks violating the capacity constraints and continue the slope-scaling procedure. The penalization proceeds by increasing the corresponding linearization factors proportionally to the flow variability ratios, i.e., $\beta_{as} = \beta_{as}(1 + \mu_s)$ and $\alpha_b = \alpha_b(1 + \mu_b)$, respectively. The sliding procedure is schematically displayed in Algorithm 3.

Algorithm 3 (Sliding repair procedure)

Require: Solution to R-ISSND;
 Define the corresponding sliding problem and solve exactly
if Solution is feasible **then**
 Define the sliding problem corresponding to the new solution and solve;
 If the resulting solution has a lower total cost, **repeat** until no further improvement is obtained.
else
 Penalize the services (and blocks) creating the infeasibility and continue the SS procedure.
end if

5.4. Ellipsoidal Search

The feasible ISSND solutions yielded either directly by addressing the approximation problem or, most often, as a result of the sliding repair procedure make up the *reference set*. We propose a new matheuristic, called *ellipsoidal search*, to intensify the search around elite solutions in this set, with a narrow focus on promising regions suggested by the history of the search and elite-solution attributes.

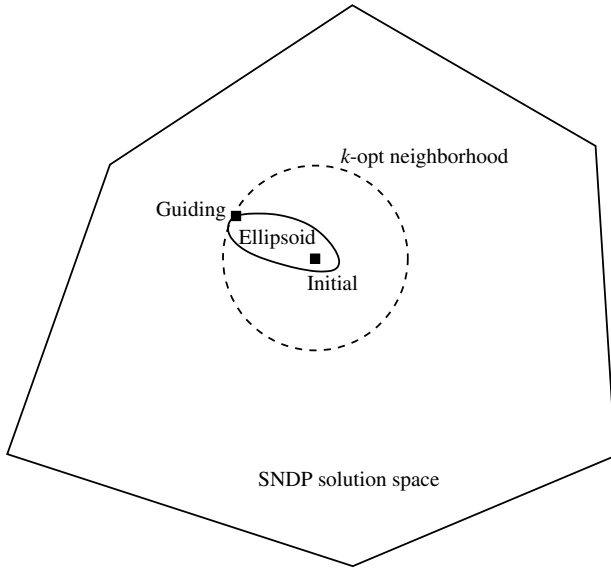
Ellipsoidal search selects two elite solutions, named *initial* and *guiding*, and uses them to define a restricted ISSND problem to be solved exactly by an MIP solver. The initial solution is then discarded from the reference set, but the pair of initial and guiding solutions is recorded in an *ellipsoidal-search memory* to avoid exploring solution regions already visited. When the MIP solver yields a solution different from the guiding solution, the new solution is added to the reference set, and the ellipsoidal search restarts; it stops otherwise.

The selection of the two elite solutions is performed based on path relinking ideas (Glover et al. 2000) as applied to the CMND by Ghamlouche et al. (2004). We thus select the guiding solution as the best solution in the reference set, and the initial solution as the solution with the maximum Hamming distance, noted k , from the guiding one.

An *elite problem* is then defined as an ISSND with a k -distance cut (or local branching cut; Fischetti and Lodi 2003) and is solved exactly starting from the initial solution. This is equivalent to implicitly exploring the k -opt neighborhood of the initial solution defined as all solutions with at most k design variables with a different status compared to the initial solution. This neighborhood includes the guiding solution; it is illustrated in Figure 8.

Given the large number of design variables proper to ISSND formulations, the k -opt neighborhood could still be quite large. We therefore further reduce the search space to a suitable ellipsoid, depicted in Figure 8, through variable

Downloaded from informs.org by [216.73.217.81] on 24 June 2026, at 01:16. For personal use only, all rights reserved.

Figure 8. Ellipsoid within the k -opt neighborhood.

selection and fixing based on the properties of the two elite solutions and the long-term search history.

All service design variables are selected for the ellipsoid definition, because each service may impact many blocks. With respect to blocks, we start with $\mathcal{B}_{\text{ini}} \cup \mathcal{B}_{\text{gui}}$, the blocks open in the initial and guiding solutions, respectively. We then add a restricted number (to keep the size of the formulation small) of other blocks with promising characteristics. The long-term memory guides this selection through the \bar{x}_b/\hat{x}_b ratio, comparing the historical average and the maximal flow on block b , which gives a measure of the block showing up steadily in the solutions of previous SS iterations. Two strategies are defined and experimentally compared in §6.2. The first selects all blocks with a ratio superior to a given threshold λ . The second considers a maximum number of blocks to add, $|\mathcal{B}_{\text{gui}}| \times \varphi$, for a parameter φ , and selects them in decreasing order of the ratio. A number of the selected variables are then fixed to preserve attributes of current elite solutions. Thus, all blocks in $\mathcal{B}_{\text{int}} \cap \mathcal{B}_{\text{gui}}$ are fixed to open (equal 1). Furthermore, flows common to both reference solutions are also fixed.

5.5. Long-Term-Memory-Based Perturbation

The goal of the perturbation phase is to take the solution method out of the current local optimum and into a different region of the search space where slope scaling can again proceed.

Perturbation is performed by increasing or reducing the linearization factors according to a *diversification-intensification* strategy based on the search information stored in the long-term memories. Diversification follows the well-known metaheuristic principle of moving the search into a not-yet-explored region of the search space. In the present context, this translates into making less/more

“interesting” a number of services (and blocks for the basic SS version) that currently are much/little used (with high/low open frequency), respectively. Intensification, on the other hand, aims to focus the search “around” good solutions, e.g., integrating components that appear consistently during previous iterations. It is thus performed by fixing services (and blocks) with high utilization (open) frequency.

To decide which blocks and services are to be involved, let \bar{f}_{ind} and σ_{ind} stand for the mean and standard deviation of the opening frequencies for services ($\text{ind} = \mathcal{S}$) and blocks ($\text{ind} = \mathcal{B}$). Define $0 \leq \theta_{\text{ind}}^+, \theta_{\text{ind}}^- \leq 1$, the thresholds indicating high and low frequency values, respectively, computed as $\theta_{\text{ind}}^+ = \bar{f}_{\text{ind}} + \omega^+ \sigma_{\text{ind}}$ and $\theta_{\text{ind}}^- = \bar{f}_{\text{ind}} - \omega^- \sigma_{\text{ind}}$, for given ω^+ and ω^- parameters. A service (block) is then said to be used frequently when $f_{\text{ind}} \geq \theta_{\text{ind}}^+$, and used rarely when $f_{\text{ind}} \leq \theta_{\text{ind}}^-$.

The rewards or penalties used to perturb the linearization factors are governed by the flow variability ratios. Diversification is performed by adding high/low penalties to services and blocks with high/low ratios, which yields

- Services: $\beta_{sa} = \beta_{sa}(1 + \mu_s)$ if $f_s \geq \theta_{\mathcal{S}}^+$, and $\beta_{sa} = \beta_{sa}\mu_s$ if $f_s < \theta_{\mathcal{S}}^-$, $a \in \mathcal{A}^{\text{SM}}$, $s \in \mathcal{S}$;
- Blocks: $\alpha_b = \alpha_b(1 + \mu_b)$ if $f_b \geq \theta_{\mathcal{B}}^+$, and $\alpha_b = \alpha_b\mu_b$ if $f_b < \theta_{\mathcal{B}}^-$, $b \in \mathcal{B}$.

Symmetrically, to intensify, high/low rewards are given to services and blocks with high/low ratios:

- Services: $\beta_{sa} = \beta_{sa}(1 - \mu_s)$ if $f_s \geq \theta_{\mathcal{S}}^+$, and $\beta_{sa} = \beta_{sa}(2 - \mu_s)$ if $f_s < \theta_{\mathcal{S}}^-$, $a \in \mathcal{A}^{\text{SM}}$, $s \in \mathcal{S}$;
- Blocks: $\alpha_b = \alpha_b(1 - \mu_b)$ if $f_b \geq \theta_{\mathcal{B}}^+$, and $\alpha_b = \alpha_b(2 - \mu_b)$ if $f_b < \theta_{\mathcal{B}}^-$, $b \in \mathcal{B}$.

Too many consecutive intensifications may focus the search on a few regions only and yield a poor exploration of the search space, whereas too many diversifications could result in continuously moving among distant solutions without a proper exploration of the regions visited. Two parameters, $I_{\text{max}}^{\text{inten}}$ and $I_{\text{max}}^{\text{diver}}$, are thus defined to guide the procedure and limit the number of consecutive intensification and diversification perturbation phases, respectively. When the new feasible solution for R-ISSND is better than (or equal to) the current best, intensification is called for, unless $I_{\text{max}}^{\text{inten}}$ consecutive intensification moves were already performed, in which case, diversification is performed. On the other hand, when that new solution is not better than the current one, diversification is called for, unless $I_{\text{max}}^{\text{diver}}$ is reached, in which case intensification is called for.

6. Computational Experiments

The performance of the proposed algorithms was computationally evaluated on a set of small to medium randomly generated instances, designed to reflect actual application settings. Algorithms were coded in C++, and experiments were run on 2.4 Hz CPU workstations with 16 GB of RAM, operating under Linux. We first describe the instances and

the generation procedure, followed by the calibration of the various algorithms. Computational results with the best set of parameters are displayed and analyzed in the last subsection.

6.1. Instance Generation

The instances were inspired by the setting of the main-line network of a major North American railroad, but without including specific details to ensure the generality of the analysis.

Instances include a description of the physical network, a number of transportation demands, and all candidate services and blocks. To enumerate candidate services, service routes are combined with possible speeds and stops. The complete service list is obtained by instantiating all promising route–stop–speed combinations at each period. Sections from the service list are then connected by transfer links and transfer-delay links in the block layer to generate potential blocks. Notice that when methods such the one we propose are used in practice, the initial set of candidate services (and blocks for the Basic_SS method) is significantly more restricted, which speeds up the computations. In such cases, planning is generally started from the transportation plan used “the last time,” enhanced with a number of potential new services reflecting changes in policy, demand, technology, regulation, and so on. For the sake of generality, we generated full potential service networks, but enforced a number of “realistic” conditions, e.g., no services and blocks with over-zigzagging routes, and no high-speed services with a high number of stops.

Two sets of 15 instances each were generated. Both sets include instances with 5–10 yards and 14–60 track sections, but differ in the number of periods, services, and blocks. Set S instances have 7 periods, between 300 and 2,674 services, and from 1,855 to 279,230 blocks; the respective figures for set L are 10, 550–3,050, and 8,900–326,550. Instance parameters are listed in electronic companion §EC.2. The figures in these tables emphasize the very large dimensions the formulation may reach in terms of service and block design variables. The numbers of flow variables and constraints also grow rapidly. To illustrate, consider that for 10 yards and 10 time periods, instances easily display over 300 million flow variables.

6.2. Calibration

Calibration aims to determine values for the major parameters of the proposed method such that it performs well over a broad range of instances. Calibration was performed on a set of random instances, called C, generated similarly to the other two, but different from them. The instances are described in electronic companion §EC.2, and detailed results supporting the decisions we report in this section are included in electronic companion §EC.3.

Three categories of parameters are important for the long-term-memory-based perturbation phase ruling when to

call on the procedure, $I_{\max}^{\text{nonimprove}}$; the number of consecutive intensifications, I_{\max}^{int} , and diversifications, I_{\max}^{div} ; and the selection range of blocks and services for perturbation, α and φ . They were fixed first with the goal of striking a balance between search length in unpromising territory (high parameter values) and moving away too soon and missing an interesting solution (low values). Three values, 8, 10, and 15, were tested for $I_{\max}^{\text{nonimprove}}$. Regarding I_{\max}^{div} and I_{\max}^{int} , setting them to 1 means alternating between intensification and diversification irrespective of possible improvement in the R-ISSND solution. Small values would provide a reasonable number of repetitions, and we tested two and three. The selection range is managed by parameters ω^+ and ω^- . We chose to keep the low-frequency thresholds equal to the average opening frequencies $\bar{n}_{\mathcal{B}}$ and $\bar{n}_{\mathcal{F}}$, which are rather small due to the large number of potential blocks and services, and set $\omega^- = 0$. Two values were tested for the high-frequency thresholds, $\omega^+ = 0$, setting the thresholds to the average opening frequencies, and $\omega^+ = 1$, which restricts the range by adding the standard deviation to the previous value.

All 12 parameter combinations were evaluated and compared over the instances of set C. The results pointed to the combination $(I_{\max}^{\text{nonimprove}}, I_{\max}^{\text{div}}, I_{\max}^{\text{int}}, \omega^+) = (10, 2, 2, 1)$, which stood out among all others and was used in all following experiments.

The next step consisted in evaluating the performance of the shortest augmenting-path heuristic in addressing the approximation problem generated during slope scaling. Experimental results indicated that, in this context, the heuristic outperforms the Simplex method, being more efficient in producing feasible solutions and enabling more slope-scaling iterations in a limited computing time. Flow distributions obtained by the augmenting path heuristic were not necessarily optimal, but they were sufficiently good to provide adequate starting points for generating better feasible ISSND solutions.

The last calibration step addressed the parameters of the ellipsoidal search. Two strategies were tested. The first, named s_1 in the following, is based on a threshold λ on the regularity of the flow on the blocks. Too small a threshold yields a large elite problem requiring a long solution time, whereas a large threshold avoids this but may also miss many interesting solutions. We compared three values for λ , 15%, 25%, and 50%. The second strategy, s_2 , limits the number of block variables to be considered in the ellipsoid through parameter φ . A value of 2 for φ generally corresponds to the $\mathcal{B}_{\text{ini}} \cup \mathcal{B}_{\text{gui}}$ set. We therefore tested 2, 5, and 10. Comparing the performance of the six settings, the second strategy outperformed the first. We therefore selected the second strategy with $\varphi = 5$, which achieved the highest score. Only a small portion of blocks are selected with this setting. For example, for instance c08, fewer than 200 block variables out of 75,000 were picked up in each elite problem.

Table 1. Performance results on instance set **S**.

Inst	CPLEX	OptGap (%)	BSS2CPLEX gap (%)	DSS2BSS gap (%)
p01	75,157	0.00	0.00	0.00
p02	72,403	0.00	0.11	0.13
p03	78,629	0.00	0.13	0.02
p04	82,784	0.00	0.00	0.00
p05	98,705	0.00	0.00	0.06
p06	107,663	2.78	−0.08	0.19
p07	256,530	35.29	−28.46	1.55
p08	227,342	36.07	−26.86	1.10
p09	167,011	26.76	−20.30	0.07
p10	228,857	36.23	−30.75	0.15
p11	140,270	10.69	−5.38	1.46
p12	187,967	17.61	−10.93	1.66
p13	×	—	—	−0.07
p14	×	—	—	−4.88
p15	×	—	—	−2.06

The ellipsoidal-search phase plays an important role in the solution procedure, as shown by the results of an initial experimental phase performed with the Basic_SS procedure (see electronic companion §EC.4). The results show that in almost all cases (29 instances of 30), including the ellipsoidal search yielded better solutions than the basic slope-scaling mechanism within the same solution time. The average improvement was 1.55% and 2.20% for the two instance sets, with a maximum improvement of over 6%.

6.3. Result Analysis

We analyze the behavior of the methods we propose by comparing the performances of the block-dynamic slope-scaling algorithm, identified as *DSS* in the following tables; the basic SS method, *BSS*; and the standard CPLEX MIP algorithm (version 12), which is also used as an MIP solver embedded into the matheuristics. All methods were allowed to run for 10 hours of CPU time, except when the maximum number of SS iterations I_{\max} was attained for the slope-scaling matheuristics. The initialization terminated after 300 iterations or five hours, whichever happened first.

Tables 1 and 2 display the computational results on instance sets **S** and **L**, respectively. CPLEX results are rounded to integer values for the sake of display clarity. The first column gives the instance name, whereas the “CPLEX” and “OptGap” columns display the objective value obtained by CPLEX and the corresponding optimality gap, respectively. Column “BSS2CPLEX gap” displays the gap of the basic slope-scaling matheuristic relative to the CPLEX solution (negative figures indicate performance improvement), whereas the last column, “DSS2BSS gap,” displays the gap of the block-dynamic SS method relative to the basic one.

The first observation is that, except for a few small instances, CPLEX was unable to find the optimal solution within the 10-hour time limit. Moreover, the efficiency

Table 2. Performance results on instance set **L**.

Inst	CPLEX	OptGap (%)	BSS2CPLEX gap (%)	DSS2BSS gap (%)
p16	94,777	1.76	−0.11	0.27
p17	90,010	2.54	−0.51	0.64
p18	128,097	0.61	0.00	0.32
p19	79,414	0.00	0.00	0.19
p20	129,305	0.00	0.00	0.00
p21	103,180	3.60	−1.48	1.51
p22	217,485	49.59	−34.58	1.26
p23	218,377	36.25	−27.58	0.36
p24	×	—	—	1.23
p25	205,420	32.00	−24.10	0.20
p26	×	—	—	1.52
p27	246,209	34.47	−26.75	1.35
p28	×	—	—	−3.60
p29	×	—	—	−1.79
p30	×	—	—	−3.08

decreased dramatically with instance size, resulting in considerable optimal gaps for medium-sized instances and the impossibility to even find a feasible solution for larger instances (e.g., p13 with 10 yards and 60 track sections).

This performance further emphasizes the impressive behavior of the proposed methods compared to CPLEX, particularly as the instance dimensions grow. For the small instances for which CPLEX found the optimum, basic slope scaling with ellipsoidal search found five optima out of seven, and solutions within 0.13% of the optimum for the others. In general, the proposed Basic_SS method obtained better solutions than CPLEX for 28 out of 30 instances tested, with a 23.57% average improvement for instances with seven yards.

The Dynamic_SS matheuristic also performed very well. It also outperformed CPLEX, both on solution quality and computing times. Compared with the basic version, the block-dynamic procedure found very close solutions with the same computing effort. Recall the additional effort required by the block-dynamic generation feature, which addresses an MIP formulation at each slope-scaling iteration. Thus, not surprisingly, the dynamic version performed somewhat less well on small and medium instances for the same computing time, with average gaps of 0.28% and 0.99% on the two instance sets. The dynamic version started to outperform the basic one when the instance dimensions grew, with improvements ranging from 0.07% for instance p13 to 4.88% for p14.

We completed the experimental analysis of the dynamic version by letting it run on a set of very large instances with 15 to 30 yards, 14 to 60 track sections, 7 time periods, and 1,000 commodities (instances are described in electronic companion §EC.2). The size of the network forbids the current implementations of CPLEX and Basic_SS to provide meaningful answers. The goal of these experiments therefore was to investigate the volume of blocks generated by the dynamic feature of the proposed matheuristic. The experiment showed that a small number of blocks is

generated by the procedure. The final number, even for the largest instances in the set, is not larger than the numbers generated a priori for the moderate-size instances of sets L and C . This indicates the robustness of the algorithm we propose with the instance dimensions.

The proposed methodology is thus efficient in providing high-quality solutions to the scheduled service network design model with double consolidation we introduced for the tactical planning of rail freight transportation. The method achieves these results in significantly less computing time than a state-of-the-art commercial MIP solver, and identifies even better ones given additional search time. Moreover, the method yields feasible solutions even when instance dimensions grow and the solver fails because of time or memory limitations.

7. Conclusions

We proposed a first comprehensive modeling framework for key tactical planning decisions for rail freight transportation systems, integrating service selection and scheduling, car classification and blocking, train make up, and routing of time-dependent customer shipments. The framework is based on a three-layer space–time network representation of the associated operations and decisions, with their space–time–activity–economic relations. It provides, in particular, a flexible and efficient approach to addressing the two-tiered consolidation nature of railways, grouping cars into blocks and blocks into trains.

Our framework, as presented in this paper, focuses on the core decisions that rail companies must make. As it stands now, a number of constraints that can be encountered in practice in rail tactical planning are not explicitly accounted for, since these depend heavily on the specific context of each railway. Adding such constraints in the framework would limit its generality and make its presentation much heavier. It is important to note, however, that many of these constraints can be dealt with quite elegantly when one generates the initial set of services and blocks by simply “filtering” services or blocks that do not satisfy them, or when dynamically generating blocks.

We also proposed a new matheuristic solution methodology, integrating exact and metaheuristic principles, to address the resulting scheduled service network design model, which takes the form of a large-size mixed-integer programming formulation. The proposed matheuristic combines slope scaling, enhanced by long-term-memory-based perturbation strategies, an effective mechanism for dynamically generating blocks, and ellipsoidal search. The latter is a new intensification mechanism we introduce to thoroughly explore a very large neighborhood of an elite solution restricted using information from the history of the search. Experimental results show the matheuristic to be efficient and yield good-quality solutions for realistically sized problem instances. The proposed methodology outperforms a state-of-the-art commercial solver, the improvements becoming increasingly spectacular with the instance dimensions.

As mentioned in the introduction, the interest of the methodological framework that we propose goes well beyond rail planning and encompasses the general field of service network design for consolidation-based transportation. It also opens interesting avenues for tackling complex multilayer design problems found in logistics, production and telecommunication applications. These applications, modeling avenues, and associated methodological challenges propose a rich research field to which we intend to contribute with our future works.

Supplemental Material

Supplemental material to this paper is available at <http://dx.doi.org/10.1287/opre.2013.1254>.

Acknowledgments

While working on this project, the first author was a doctoral student with the Département d’informatique et de recherche opérationnelle, Université de Montréal, and the second author was the Natural Sciences and Engineering Council of Canada (NSERC) Industrial Research Chair on Logistics Management, and an adjunct professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway. Funding for this project was provided by the NSERC through its Industrial Research Chair and Discovery Grants programs; by the partners of the chair, CN, Rona, Alimentation Couche-Tard, and the Ministry of Transportation of Québec; and by the Fonds de recherche du Québec—Nature et technologies.

References

- Ahuja RK, Cunha CB, Sahin G (2005) Network models in railroad planning and scheduling. *Tutorials in Operations Research* (INFORMS, Catonsville, MD), 54–101.
- Ahuja RK, Jha KC, Liu J (2007) Solving real-life railroad blocking problems. *Interfaces* 37(5):404–419.
- Andersen J, Crainic TG, Christiansen M (2009a) Service network design with asset management: Formulations and comparative analyses. *Transportation Res. C* 17(2):197–207.
- Andersen J, Crainic TG, Christiansen M (2009b) Service network design with management and coordination of multiple fleets. *Eur. J. Oper. Res.* 193(2):377–389.
- Assad AA (1980) Models of rail networks: Toward a routing/make-up model. *Transportation Res. B* 14(1–2):101–114.
- Barnhart C, Jin H, Vance PH (2000) Railroad blocking: A network design application. *Oper. Res.* 48(4):603–614.
- Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.
- Bodin LD, Golden BL, Schuster AD, Romig W (1980) A model for the blocking of trains. *Transportation Res. B* 14(1–2):115–120.
- Brännlund U, Lindberg PO, Nou A, Nilsson J-E (1998) Railway timetabling using Lagrangian relaxation. *Transportation Sci.* 32(4):358–369.
- Caprara A, Fischetti M, Toth P (2002) Modeling and solving the train timetabling problem. *Oper. Res.* 50(5):851–861.
- Caprara A, Monaci M, Toth P, Guida PL (2006) A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Appl. Math.* 154:738–753.
- Cordeau J-F, Toth P, Vigo D (1998) A survey of optimization models for train routing and scheduling. *Transportation Sci.* 32(4):380–404.

- Crainic TG (2000) Service network design in freight transportation. *Eur. J. Oper. Res.* 122:272–288.
- Crainic TG, Kim KH (2007) Intermodal transportation. Barnhart C, Laporte G, eds. *Transportation, Handbooks in Operations Research and Management Science*, Vol. 14, Chap. 8 (North-Holland, Amsterdam), 467–537.
- Crainic TG, Rousseau JM (1986) Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Res. B* 20(3):225–242.
- Crainic TG, Ferland JA, Rousseau JM (1984) A tactical planning model for rail freight transportation. *Transportation Sci.* 18(2):165–184.
- Crainic TG, Gendron B, Hernu G (2004) A slope scaling/Lagrangian perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *J. Heuristics* 10(5):525–545.
- Fischetti M, Lodi A (2003) Local branching. *Math. Programming, Ser. B* 98:23–47.
- Ghamlouche I, Crainic TG, Gendreau M (2003) Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Oper. Res.* 51(4):655–667.
- Ghamlouche I, Crainic TG, Gendreau M (2004) Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Ann. Oper. Res.* 131(1–4):109–133.
- Glover F, Laguna M, Marti R (2000) Fundamentals of scatter search and path relinking. *Control and Cybernetics* 29(3):653–684.
- Goossens JW, van Hoesel S, Kroon L (2004) A branch-and-cut approach for solving railway line-planning problems. *Transportation Sci.* 38(3):379–393.
- Gorman MF (1998a) An application of genetic and tabu searches to the freight railroad operating plan problem. *Ann. Oper. Res.* 78:51–69.
- Gorman MF (1998b) Santa Fe railway uses an operating plan model to improve its service design. *Interfaces* 28(4):1–12.
- Haghani AE (1989) Formulation and solution of combined train routing and makeup, and empty car distribution model. *Transportation Res. B* 23(6):433–452.
- Hewitt M, Nemhauser GL, Savelsbergh MWP (2010) Combining exact and heuristic approaches for the capacitated fixed charge network flow problem. *INFORMS J. Comput.* 22(2):314–325.
- Huntley CL, Brown DE, Sappington DE, Markowicz BP (1995) Freight routing and scheduling at CSX Transportation. *Interfaces* 25(3):58–71.
- Ireland P, Case R, Fallis J, Van Dyke C, Kuehn J, Meketon M (2004) The Canadian Pacific Railway transforms operations by using models to develop its operating plans. *Interfaces* 34(1):5–14.
- Keaton MH (1989) Designing optimal railroad operating plans: Lagrangian relaxation and heuristic approaches. *Transportation Res.* 23B:415–431.
- Keaton MH (1992) Designing railroad operating plans: A dual adjustment method for implementing Lagrangian relaxation. *Transportation Sci.* 26(4):263–279.
- Kim D, Pardalos PM (1999) A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Oper. Res. Lett.* 24:195–203.
- Kim D, Pan X, Pardalos PM (2006) An enhanced dynamic slope scaling procedure with tabu scheme for fixed charge network flow problem. *Comput. Econom.* 27:273–293.
- Magnanti TL, Wong RT (1984) Network design and transportation planning: Models and algorithms. *Transportation Sci.* 18(1):1–55.
- Marín A, Salmerón J (1996) Tactical design of rail freight network. Part I: Exact and heuristic methods. *Eur. J. Oper. Res.* 90(1):26–44.
- Morlok EK, Peterson RB (1970) Final report on a development of a geographic transportation network generation and evaluation model. *Transportation Research Forum* 11:71–105.
- Newman AM, Yano CA (2000) Centralized and decentralized train scheduling for intermodal operations. *IIE Trans.* 32:743–754.
- Newton HN, Barnhart C, Vance PH (1998) Constructing railroad blocking plans to minimize handling costs. *Transportation Sci.* 32(4):330–345.
- Pedersen MB, Crainic TG (2007) Optimization of intermodal freight service schedules on train canals. Research Report CIRRELT-2007-51, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montréal, Quebec, Canada.
- Pedersen MB, Crainic TG, Madsen OBG (2009) Models and tabu search meta-heuristics for service network design with asset-balance requirements. *Transportation Sci.* 43(2):158–177.
- Zhu E (2011) Scheduled service network design for integrated planning of rail freight transportation. Publication CIRRELT-2011-11, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montréal, Quebec, Canada.
- Zhu E, Crainic TG, Gendreau M (2009) Integrated service network design for rail freight transportation. Publication CIRRELT-2009-45, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montréal, Quebec, Canada.

Endong Zhu earned his Ph.D. in operations research at Université de Montréal, with research interests primarily in transportation science and logistics. His research work focused on the planning and scheduling of railway transportation services. He was the first place award winner of the student paper contest of the Railway Application Section of INFORMS in 2009. He works in China as a senior data scientist and operations research specialist.

Teodor Gabriel Crainic is a professor of operations research, transportation, and logistics at the School of Management, Université du Québec Montréal, where he holds the Logistics Management Chair. His main interest areas are in network, integer, and combinatorial optimization, metaheuristics, and parallel computing applied to the planning and management of complex systems, particularly in transportation and logistics. He received the 2006 Merit Award of the Canadian Operational Research Society and has been elected Member of the Royal Society of Canada: The Academies of Arts, Humanities, and Sciences of Canada.

Michel Gendreau is a professor of operations research at the Department of Applied Mathematics and Industrial Engineering of École Polytechnique de Montréal (Canada), where he holds the NSERC/Hydro-Québec Chair on the Stochastic Optimization of Electricity Generation. His main research area is the application of operations research methods to energy and transportation systems planning and operation. In 2001 he received the Merit Award of the Canadian Operational Research Society in recognition of his contributions to the development of operations research in Canada. He was elected a Fellow of INFORMS in 2010.