Received: 18 July 2020 | Revised: 12 March 2021 | Accepted: 12 April 2021 | Published on: 6 May 2021

DOI: 10.1002/net.22042

SPECIAL ISSUE ARTICLE

WILEY

An innovative operation-time-space network for solving different logistic problems with capacity and time constraints

Daniela Ambrosino | **Veronica Asta**

Department of Economics and Business Studies, University of Genoa, and Research Center on Logistics, Transport and Infrastructures - CIELI, Genoa, Italy

Correspondence
Daniela Ambrosino, Department of Economics and Business Studies, University of Genoa, and Research Center on Logistics, Transport and Infrastructures - CIELI, Genoa, Italy.
Email: daniela.ambrosino@economia.unige.it

Abstract

This work describes an innovative operation-time-space network that can be easily adapted to model different logistic problems involving time and space decisions. In the recent literature, different time-space networks have been proposed to deal with specific requirements of problems arising in the logistic and transportation fields. The proposed network is able to deal with different types of capacity and time constraints that characterized the most part of logistic problems. The article explains how to construct the operation-time-space network, how to model time and capacity constraints of different types; moreover, it presents an application of this network to solve a logistic problem arising in the rail-sea exchange node, that is the scheduling of port rail shunting operations. Other applications are briefly introduced. Finally, the well-known quay crane scheduling problem is modeled, thanks to the operation-time-space network; the related flow model is tested by using a benchmark suite available in the literature.

KEYWORDS

capacity constraints, example of logistic applications, logistic problems in rail-sea terminals, network flow model, time constraints, time-space network

1 | INTRODUCTION AND LITERATURE REVIEW

Time-space networks are generally used for solving problems containing both time and space decisions. When temporal aspects are relevant, a physical network can be extended to a time-space network (TSN). TSN has been proposed for solving problems in different fields, in particular in logistic and transportation systems fields. In the recent literature, even more structured TSNs have been proposed for representing and solving complex problems, as briefly discussed in the following.

In the logistic field, the activities are related to plan, organize, and control materials, goods, and information flows, often in an integrated way. At strategic level, some network design problems, including location, inventory, and routing decisions, may arise, while at operative level operational problems emerging in a single node of the logistic network can be considered: from procurement, production to distribution, and transportation problems. For what concerns the transportation systems, the activities are related to the design and the management of a network to facilitate the movements of passengers, freight, and again information, thus involving both strategical and operational decisions. The mobility is supported by one or more transportation modes, such as road, rail, maritime, and air transport.

TSN for logistic problems

In the logistic field, TSNs have been used for solving, just to cite some, service network design, crane scheduling, and vehicle routing problems.

An example of service network design is in [1]. They use a time-space network to represent a ship scheduling and containerized cargo routing problem formulating it as a multicommodity flow problem with side constraints. Each node of the network represents a port on a day of a week, that is the considered planning horizon. Different types of arcs are used to represent the ships in the ports, their movements, and the movements of cargo of the ships.

At operational level, a service problem is presented in [4]; the authors use a time-space network for solving the air transportation freight forwarder service problem. A freight forwarder has to organize shipments, during a certain horizon, by choosing the best options within a wide offer of transportation services. Costs must be minimized and delivery times respected. The possible sequences of transportation services available for a shipment, characterized by an origin, and a destination, are the layers of the network used by the authors. A flow model is solved on the network.

Scheduling problems, which contain both time and space decisions, are often faced by TSNs. In [5] the authors develop a time-space network flow formulation with noncrossing constraints for the crane scheduling problem for a vessel during terminal operations.

In [19], the authors use an event-based time-space network flow model to solve a crane scheduling problem in a coil warehouse. They determine, simultaneously, the sequence of the crane operations (i.e., storage, retrieval, and shuffling requests) and the positions to which the coils are moved. In the event-based time-space network, each node represents a location in the warehouse at a specific time of the planning horizon, that is the end of a specific scheduling stage; each edge indicates a move of a crane between two locations in a stage. A stage corresponds to a move of a crane.

The most space-time networks are classified as discrete-time approaches in which the planning horizon is divided into a number of time units of uniform duration. In [19], the authors prefer a different approach since the decision activities only happen at the start and end point of each movement of the crane, rather than at every time point in the planning horizon. Thus, they use an event-based time-space network flow formulation based on continuous time.

In a recent work [15], a time-space network is used to model the an Automated guided vehicles (AGV) routing problem formulating it as a mixed-integer linear programming problem. The system is described in detail, with its machines, its products, and the tasks. Each node of the network represents a machine. A task is picked up, delivered, and then processed by a machine. Finally, the task changes into another one. Each product is finished via few tasks and then conveyed to the depository. The author considers capacity for the machines and time constraints on the pickup-and-delivery.

Mahmoudi and Zhou [13] propose a three-dimensional state-space-time hypernetwork to formulate vehicle routing problem with pickup and delivery services with time windows (VRPPDTW) as a time-discretized, multidimensional, multicommodity flow model with linear objective function and constraints. With this novel network, the authors are able to enumerate possible transportation states at any given time along vehicle space-time paths and to forward a dynamic programming solution algorithm to solve the single VRPPDTW problem.

Lu et al. [12] propose a resource-space-time network to formulate a resource constrained location routing problem as a multicommodity flow problem. The proposed network is a three-dimensional network that combines the space dimension, the time dimension, and the resource states. Thus, a node of the physical network in new three-dimensional network indicates that a vehicle in that physical node maintains a certain resource level at a given instant time. The authors consider both transportation demands and resources recharging requirements to satisfy. The arcs of the network correspond to a vehicle traveling from one physical location to another one, requiring a certain time and a certain resource.

TSN for transportation systems

The problems arising in the field of transportation planning and management have been faced by many researchers proposing specific time-space networks to respond to particular requirements of the different transportation systems; among others we can cite [8] for road network with signal settings, [9] for urban transit network, and [11] for bike-sharing network.

More innovative applications of TSNs are proposed in [17] to maximize the individual accessibility within travel time budgets, in [10] for studying the observability problem in public transportation systems, and in [18] to both determine the last-train timetable on each line of the urban rail transit network and evaluate the schedule performances in terms of quality of the different trains connection.

Often a physical network on a plane is transformed into a space-time network represented by three dimensions [18]: two coordinates denote the space and the third denotes the discretized time horizon. In this way, each point of the network is used to represent the specific position of a mean of transport or a passenger in a particular instant time. Moreover, it is possible to represent the movements of both the means of transport and passengers, thanks to different kind of arcs: space-time running arcs, space-time dwelling arcs, space-time transfer arcs, and space-time waiting arcs.

Other applications of TSNs

Despite of the above-mentioned uses of the time-space networks in their different forms, some authors adopt them also for facing other problems, related again to the logistic field. Among these innovative uses of networks we can mention [20] and [3] that use a TSN for evaluating a truck appointment system for a maritime terminal. Zehendner and Feillet [20]

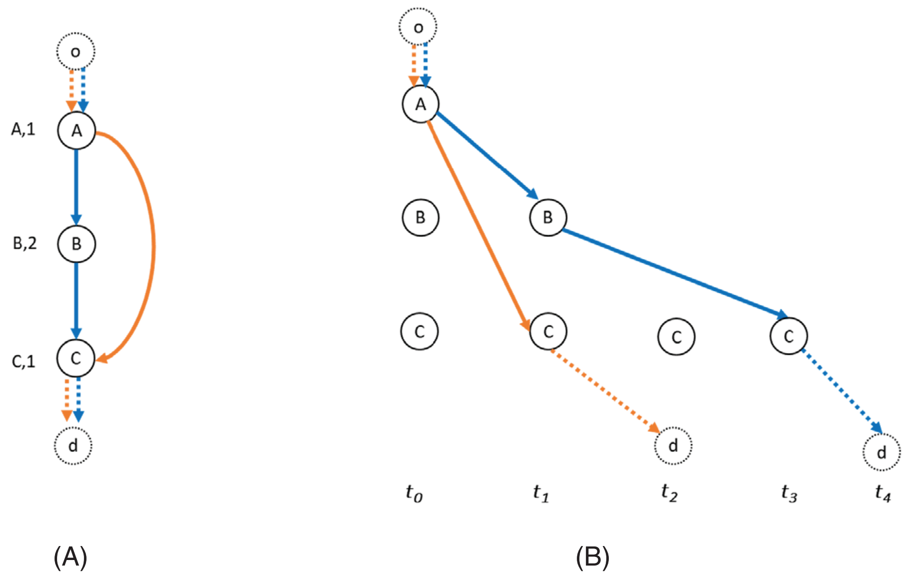

FIGURE 1 Physical network (A) and time-space network (B)

determine the number of truck appointments to offer while allocating straddle carriers to different transport modes by solving a mixed-integer linear programming model based on the network; Ambrosino and Peirano [3] investigate the management of truck arrivals by offering a nonmandatory truck appointment system; the authors propose a multicommodity network flow model for representing a general terminal in order to manage the truck arrivals and to grant the trucks a certain service level.

From the above brief discussion, we can note that different TSNs have been proposed to deal with specific requirements of problems having in common spacial and time decisions. The aim of this work is to generalize a TSN in such a way to be able to use it for solving different problems in the logistic field, both at strategical and operational level. In particular, we try to generalize the network proposed to solve a scheduling problem arising in the maritime rail-sea exchange nodes [2], that is the scheduling of port rail shunting operations (PRSSP). Our first aim is to be able to solve the same optimization problem for different operators that can have to manage different jobs, different resources on different infrastructures and, then, to solve different flow optimization models for defining the scheduling of several activities sharing some limited resources and having some time constraints. Moreover, we suggest some other uses to face strategic decisions for network design problems in maritime logistics and, finally for solving other logistic problems that concern decisions involving time and space aspects.

The remainder of this article is organized as follows. Section 2 illustrates how to construct the innovative operation-time-space network. Sections 3 shows the usage of this network to face a maritime logistic problem, the PRSSP. Finally, Section 4 reports other possible applications of the proposed network and some conclusions.

2 | THE INNOVATIVE OPERATION-TIME-SPACE NETWORK

In this section, the innovative operation-time-space network is presented. Let us introduce it by using an example.

Consider a network representing the physical layout of a plant where three activities (A, B, and C) can be performed, that is, the physical network reported in Figure 1(A), in which two paths, related to two jobs, are shown. Each path indicates the sequence of activities executed by each job. The weight of each node represents the duration of the activity.

Researchers often find more convenient to use a time-space network to follow the execution of activities during time, as reported in Figure 1(B). The paths indicate the sequence of activities executed by each job, but the time dimension is here more simple to understand. Time horizon T is discretized; T is split into equal time intervals $T = \{t_0, t_1, t_2, \dots, t_s\}$.

In the following, we propose an innovative operation-time-space network that can be easily derived by the time-space network of Figure 1(B) and that permits to know in each time period the activities that the jobs are executing and the resources consumption.

In the proposed network there are:

- horizontal arcs: arcs that represent the execution, during the time horizon, of the activities;
- vertical arcs (also called transfer arcs): arcs that permit the jobs to pass from one activity to another; transfer arcs are present for each couple of compatible activities, that is, couple of activities that can be realized in sequence.

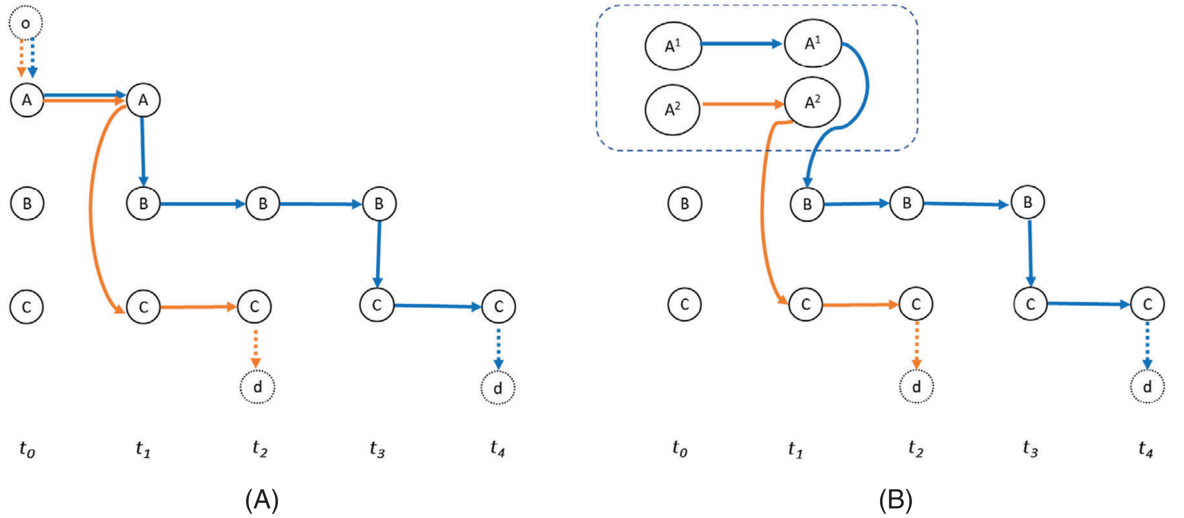

FIGURE 2 Operation-time-space networks

This network is depicted in Figure 2(A), where the two paths of the jobs corresponding to those reported in Figure 1 are shown.

Each time interval t in T is here denoted $[t; t+1)$ and when an operation is related to time t (i.e., has index t) means that it happens in the interval $[t; t+1)$. The blue job in Figure 2(A) enters in the network in t_0 and immediately starts activity A. The blue job finishes activity A in t_1 and passes to activity B which starts in t_1. Activity B is performed until t_3; then, the job starts activity C. After having completed C, the job in t_4 leaves the network. The readers can note that the nodes of the network represent the activities but, thanks to the vertical arcs, nodes also indicate the ending and the starting time of the corresponding activities.

This operation-time-space network allows an easier implementation of capacity constraints that may regard either an activity, a set of activities, or the whole network. Moreover, different levels of detail are allowed; for example, we can include details on the resources required to perform the activities.

Looking at Figure 2(A), we can note that activity A has a capacity to perform the both jobs in t_0. The resources needed to perform activity A are not explicitly considered in the network. Now, suppose that it is required to distinguish among resources available for performing activity A. In this case we can duplicate the node A, one node A for each resource, as in Figure 2(B) (where two resources are considered).

Now, let us suppose to have a new job, the green one in Figure 3, that has to pass through B and C. We can distinguish different cases:

i) B can operate two items in each t (and the resources used in B are not distinguished); the paths on the network in Figure 3(A) represent this new situation.
ii) B has the capacity to execute only one job in each t; thus, the blue job cannot start activity B before t_2. Two different situations may arise:
   iia) the job cannot wait in the network, that is each job has to execute its activities in sequence. In this case, the blue job has to execute activity B as soon as it finishes activity A, and for doing that the blue job has to enter in the network later, in t_1; then, it executes A and it starts B in t_2, as depicted in Figure 3(B).
   iib) the job can wait in the network, that is there is a waiting area/buffer (W). In this case, the blue job finishes activity A and has to wait before starting activity B. The blue job enters in the buffer in t_1, remains in W and then in t_2 passes to activity B, as depicted in Figure 4.

This operation-time-space network also allows an easier implementation of time constraints. In addition to precedence relationships that are represented in the network by vertical arcs, a job can have time constraints indicating the exact starting and/or ending time for an activity that it has to execute, either a deadline or a time window for starting and/or ending an activity. These constraints are discussed in detail in [6] where the authors developed a method to find the critical path in an activity network with different types of time constraints on activities.

We will show how this operation-time-space network permits to consider all these time constraints in Sections 2.1 and 3.

Different optimization problems can be solved as 0/1 integer linear flow models based on the discretized operation-time-space network described above. Let us introduce more formally this network.

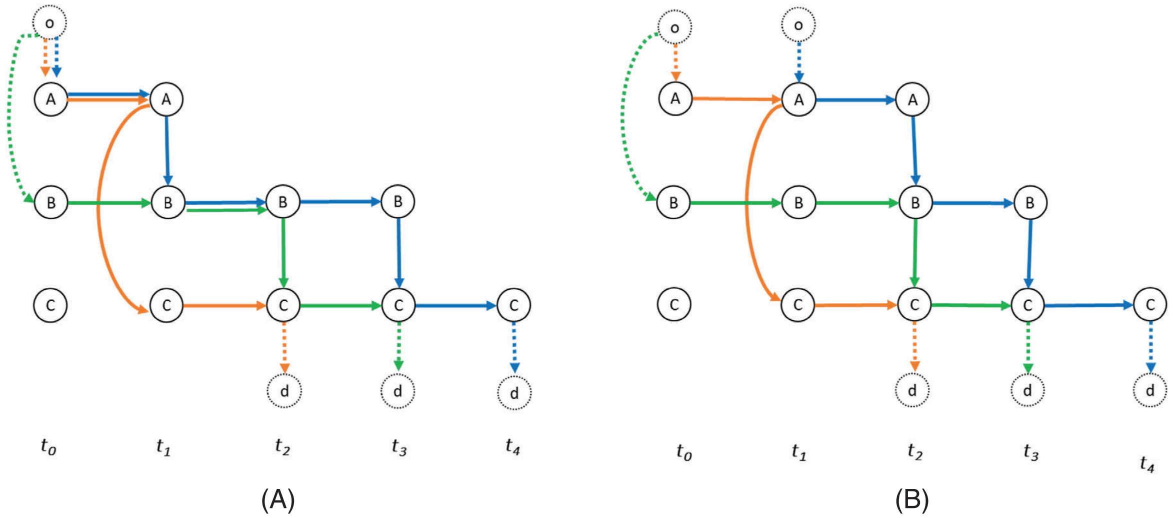

FIGURE 3 Operation-time-space networks without wait buffers

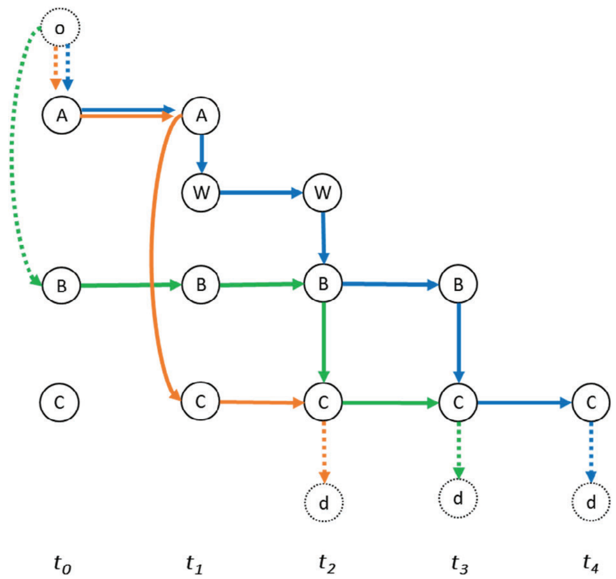

FIGURE 4 Operation-time-space network with wait buffers

Let $G = (N, A)$ be the operation-time-space network, where N is the set of nodes and A the set of arcs. Note that, N is the union of different subsets that have to be better specified; their specification depends on the presence of one or more resources that we need to model separately for each activity (as shown in Figure 2). Moreover, N includes also the source and the sink nodes of the network.

Denote by the following

O the set of activities that can be performed and that we want to model on G, included also the entry and the exit from the network, and the waiting buffers if they are available;

J the set of jobs that need to perform either all activities in O or some of the activities in O;

O_j the set of activities that job j must perform;

O^+ the set of activities for which we have to distinguish the available resources;

$O^- = O - O^+$ the set of activities for which we have not to distinguish the available resources;

O^C the set of couple of activities that can be executed in sequence, necessary to define transfer arcs (that permit the jobs to pass from one activity to another and to enter and leave the network);

$\mathcal{R}_i$ the set of resources available for executing the activity i, $i \in O^+$;

H the set of groups of activities sharing a given resource;

O_h the set of activities of group h, $h \in H$.

$\mathcal{N} = \cup_{i \in O} \mathcal{N}^i$ where, depending on the necessity to distinguish the set of operations in O^- and O^+ :

$\mathcal{N}^i = \{n_t^i | t \in T\}, \forall i \in O^-$ the set of nodes representing the activity i, in each t of the time horizon;
$\mathcal{N}^i = \cup_{r \in \mathcal{R}_i} \mathcal{N}_r^i, \forall i \in O^+$ with
$\mathcal{N}_r^i = \{n_{r,t}^i | t \in T\}$ the set of nodes representing the resource r of activity i, $i \in O^+$, $r \in \mathcal{R}_i$, in each t of the time horizon.
$\mathcal{A} = (\cup_{i \in O} \mathcal{A}^{\mathcal{N}^i}) \cup (\cup_{i,j \in O^C} \mathcal{A}^{\mathcal{N}^i \mathcal{N}^j})$ the set of arcs of the network, given by the union of horizontal arcs ($\mathcal{A}^{\mathcal{N}^i}$) and vertical ones ($\mathcal{A}^{\mathcal{N}^i \mathcal{N}^j}$), defined as in the following.

As before, each subset of the horizontal arcs $\mathcal{A}^{\mathcal{N}^i}$ can be defined in one of the following ways, depending on the kind of activity it refers to:

$\mathcal{A}^{\mathcal{N}^i}$ the set of arcs $\{(n_t^i, n_{t+1}^i), \forall t \in T\}, \forall i \in O^-$
$\mathcal{A}^{\mathcal{N}^i} = \cup_{r \in \mathcal{R}_i} \mathcal{A}^{\mathcal{N}_r^i}, \forall i \in O^+$ with
$\mathcal{A}^{\mathcal{N}_r^i}$ the set of arcs $\{(n_{r,t}^i, n_{r,t+1}^i), \forall t \in T\}, \forall i \in O^+, \forall r \in \mathcal{R}_i$.

Note that $\mathcal{A}_t^{\mathcal{N}^i}$ and $\mathcal{A}_t^{\mathcal{N}_r^i}$ represent the arcs (n_t^i, n_{t+1}^i) and $(n_{r,t}^i, n_{r,t+1}^i)$, respectively.
For what concerns vertical arcs, they link couples of activities that can be executed in sequence. Some different definitions are required, depending on the kind of activity they refer to:

$\mathcal{A}^{\mathcal{N}^i \mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_t^l), \forall t \in T\}, \forall i, l \in O^C : i, l \in O^-$
$\mathcal{A}^{\mathcal{N}^i \mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_{r,t}^l), \forall t \in T, \forall r \in \mathcal{R}_l\}, \forall i, l \in O^C : i \in O^-, l \in O^+$
$\mathcal{A}^{\mathcal{N}^i \mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_t^l), \forall t \in T, \forall r \in \mathcal{R}_i\}, \forall i, l \in O^C : i \in O^+, l \in O^-$
$\mathcal{A}^{\mathcal{N}^i \mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_{r',t}^l), \forall t \in T, \forall r \in \mathcal{R}_i, \forall r' \in \mathcal{R}_l\}, \forall i, l \in O^C : i \in O^+, l \in O^+$.

Here we denote:

$\mathcal{A}_t^{\mathcal{N}^i+}$ ($\mathcal{A}_t^{\mathcal{N}_r^i+}$) the subset of outbound vertical arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible activities;
$\mathcal{A}_t^{-\mathcal{N}^i}$ ($\mathcal{A}_t^{-\mathcal{N}_r^i}$) the subset of inbound vertical arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible activities.

Finally, let us denote $\mathcal{A}_t^+$ the set of outbound arcs in t, $t \in T$.

2.1 | The network flow model

In this section, we introduce a flow formulation based on the network described above that can be adapted to solve many real applications.
The following additional notation useful for the flow formulation is introduced.

d_{ij} the duration of activity i for job j, $\forall j \in J, \forall i \in O_j$;
s_{ij} the starting time of activity i for job j, $\forall j \in J, \forall i \in O_j$;
e_{ij} the ending time of activity i for job j, $\forall j \in J, \forall i \in O_j$;
$[s_{ij}^{\min}, s_{ij}^{\max}]$ the time window for job j for starting the execution of activity i, $\forall j \in J, \forall i \in O_j$;
$[e_{ij}^{\min}, e_{ij}^{\max}]$ the time window for job j for completing activity i, $\forall j \in J, \forall i \in O_j$;
u_t^i the maximum number of jobs that can execute activity i in t, $\forall i \in O^-, \forall t \in T$;
k_t^h the maximum number of jobs that can execute activities of group h in t, $\forall h \in H, \forall t \in T$;
q_t the maximum number of jobs that in each t can be present in the network, $\forall t \in T$.

Decision variables
The model based on the network described above has the following flow variables indicating for each job, when it is performing the operations (horizontal arcs) and when it is passing from one operation to another (vertical arcs):

$x_{a,j} \in \{0, 1\} \forall j \in J, \forall a \in \mathcal{A}$,
$x_{a,j} = 1$ if arc a is used for job j, 0 otherwise.

Constraints
Classical flow conservation constraints, capacity and time constraints are presented here below. Note that, the presence of activities (and thus of nodes), that are split for specifying the available resources, affects the readability of both the notation and the constraints here reported, but permits a real representation of particular capacity constraints.

Flow conservation constraints
The flow conservation constraints assure, for each unit of flow on the network, that is, for each job j, that the sum of all the variables related to the inbound arcs of a node in time instant t is equal to the sum of the variables related to the outbound arcs of the same node.

$$\sum_{a : a \in \mathcal{A}_{t-1}^{\mathcal{N}^i} \cup \mathcal{A}_t^{-\mathcal{N}^i}} x_{a,j} = \sum_{a : a \in \mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i+}} x_{a,j} \quad \forall j \in J, \forall i \in O_j, \forall t \in T$$

Time constraints

The constraints related to time requirements can be easily written specifying in the sum of the constraints what are the time instants t to include.

For example, if it is required a job j to start activity i within a time window $[s_{ij}^{\min}, s_{ij}^{\max}]$, we can simply impose to select one arc among the subset $\mathcal{A}_t^{-\mathcal{N}^i}$ of inbound vertical arcs of node representing the activity i, with t such that $s_{ij}^{\min} \le t \le s_{ij}^{\max}$. The resulting constraint is the following:

$$\text{(TC1)} \quad \textstyle\sum_{a:a \in (\cup_{s_{ij}^{\min} \le t \le s_{ij}^{\max}} \mathcal{A}_t^{-\mathcal{N}^i})} \; x_{a,j} = 1.$$

If the requirement for job j is to end activity i within a time window, we can simply rewrite the above constraint requiring to select one arc among the subset $\mathcal{A}_t^{\mathcal{N}^i+}$ of outbound vertical arcs of node representing the activity i, with t within the fixed time window.

$$\text{(TC1}'\text{)} \quad \textstyle\sum_{a:a \in (\cup_{e_{ij}^{\min} \le t \le e_{ij}^{\max}} \mathcal{A}_t^{\mathcal{N}^i+})} \; x_{a,j} = 1.$$

If the requirement for job j is to start activity i within a deadline s_{ij}, we can simply impose to select one arc among the subset $\mathcal{A}_t^{-\mathcal{N}^i}$ of inbound vertical arcs of node representing the activity i, with t within the fixed time, that is, $t \le s_{ij}$ The resulting constraint is the following:

$$\text{(TC2)} \quad \textstyle\sum_{a:a \in (\cup_{t \le s_{ij}} \mathcal{A}_t^{-\mathcal{N}^i})} \; x_{a,j} = 1.$$

If the requirement for job j is to end activity i within a deadline e_{ij}, we can simply rewrite the above constraint as the following:

$$\text{(TC2}'\text{)} \quad \textstyle\sum_{a:a \in (\cup_{t \le e_{ij}} \mathcal{A}_t^{\mathcal{N}^i+})} \; x_{a,j} = 1.$$

If job j must complete all its activities within a time limit (i.e., it has to leave the network within a time limit), having considered the exit of the job j as an operations in O_j, we can use the above constraints formulation.

Finally, if job j has to start the activity i in a precise instant s_{ij}, the required constraint is the following:

$$\text{(TC3)} \quad \textstyle\sum_{a:a \in \mathcal{A}_{t=s_{ij}}^{-\mathcal{N}^i}} \; x_{a,j} = 1.$$

While if job j has to complete activity i in a precise instant e_{ij}, we impose:

$$\text{(TC3}'\text{)} \quad \textstyle\sum_{a:a \in \mathcal{A}_{t=e_{ij}}^{\mathcal{N}^i+}} \; x_{a,j} = 1.$$

Capacity constraints for each activity/for a group of activities/for the whole network

If there is a limitation on the number of jobs that can execute simultaneously an activity i in each t, we need some constraints imposing that the number of outbound (horizontal and vertical) arcs of node representing the activity i is less or equal to this limit (i.e., u_t^i):

$$\text{(CC1)} \quad \textstyle\sum_{j \in J} \sum_{a:a \in \mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i+}} \; x_{a,j} \le u_t^i \qquad \forall t \in T, \forall i \in O^-.$$

For activities with specific resources, that is, $i \in O^+$, we need the following constraints requiring the number of outbound (horizontal and vertical) arcs of node representing the resource r of activity i is less or equal to one for each t:

$$\text{(CC2)} \quad \textstyle\sum_{j \in J} \sum_{a:a \in \mathcal{A}_t^{\mathcal{N}_r^i} \cup \mathcal{A}_t^{\mathcal{N}_r^i+}} \; x_{a,j} \le 1 \qquad \forall t \in T, \forall i \in O^+, \forall r \in \mathcal{R}_i.$$

At most k_t^h jobs can execute simultaneously activities of group h in each t. Therefore, the number of outbound (horizontal and vertical) arcs of nodes representing the activities i belonging to the group h (O_h) must be less or equal to k_t^h for each t:

$$\text{(CC3)} \quad \textstyle\sum_{j \in J} \sum_{a:a \in \cup_{i \in O_h} (\mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i+})} \; x_{a,j} \le k_t^h \qquad \forall h \in H, \forall t \in T.$$

If there is a maximum number of activities that can be executed simultaneously in each t, for example, due to the number of available resources shared among all the activities, the required constraints are the following:

$$\text{(CC4)} \quad \textstyle\sum_{j \in J} \sum_{a:a \in \mathcal{A}_t^+} \; x_{a,j} \le q_t \qquad \forall t \in T.$$

Processing time of the activities

The time that job j has to spend for executing activity i is known (i.e., d_{ij}); thus, the following constraints are defined:

$$\text{(PTC)} \quad \textstyle\sum_{a \in \mathcal{A}^{\mathcal{N}^i}} \; x_{a,j} = d_{ij} \qquad \forall j \in J, \forall i \in O_j.$$

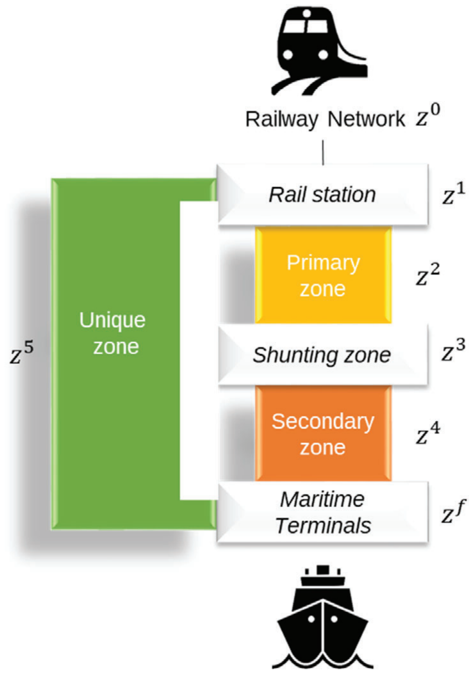

FIGURE 5 The rail-sea modality exchange node configuration under investigation

Objective function

The flow optimization models used for solving logistic problems present different objective functions, among others the maximization of the flow, the minimization of the total costs of the flow passing through the network. In scheduling problems the minimization of the total time required to execute all the jobs is a common objective. In time constrained networks different aims can be perceived, like the minimization of the total waiting time spent by the jobs in the waiting areas (buffers).

2.2 | Reducing the complexity of the networks and the models size

The network flow models generally present huge dimensions due to the high number of variables to generate, that is, one variable for each job and each arc of the network. Different strategies to reduce the complexity and to speed up the computation have been proposed by many authors. Just to cite some examples, strategies for reducing the model size that are based on the identification of redundant variables are presented in [19]; Mahmoudi and Zhou [13] define rational rules that permit them to delete some vertex (some arcs) in the network, thus reducing the network size; valid inequalities to reduce the required computational time for solving the flow model are proposed in [15].

Referring to the proposed network, we can try to reduce the number of variables of the flow model by exploiting the restrictions that characterized each job (time windows, deadlines, due dates). Moreover, we think that it is possible the reduce the computational time required to solve a problem only deeply analyzing the problem characteristics behind the TSN used to model it. In fact, all the above-mentioned strategies are strongly dependent of the characteristics of the problem under investigation.

3 | A CASE STUDY: SCHEDULING OF PORT RAIL SHUNTING OPERATIONS

In this section, we show how it is possible to use the operation-time-space network proposed in this article for the scheduling of port rail shunting operations [2]. Shunting operations permit to transfer trains or portions of them within the port area.

The port area generally includes one railway station, a shunting zone, and several maritime terminals, as depicted in Figure 5.

Let us describe the physical paths that an export and an import train can follow in the port area, supposing to have only one shunting park. An export train arrives and waits (if necessary) in the rail station before going to its maritime terminal through either an unique shunting operation, or by performing two operations passing through the shunting park. An import train performs the same opposite paths. A train that performs an unique shunting operation uses a track connecting directly the station and its maritime terminal. This operation is realized in a zone here called unique zone, the green one in Figure 5. A train that passes through the shunting park has to perform two transfer operations; it has to use a track connecting the station to the shunting park, to wait (if necessary) in a track of the shunting park, and to use a track connecting the shunting park

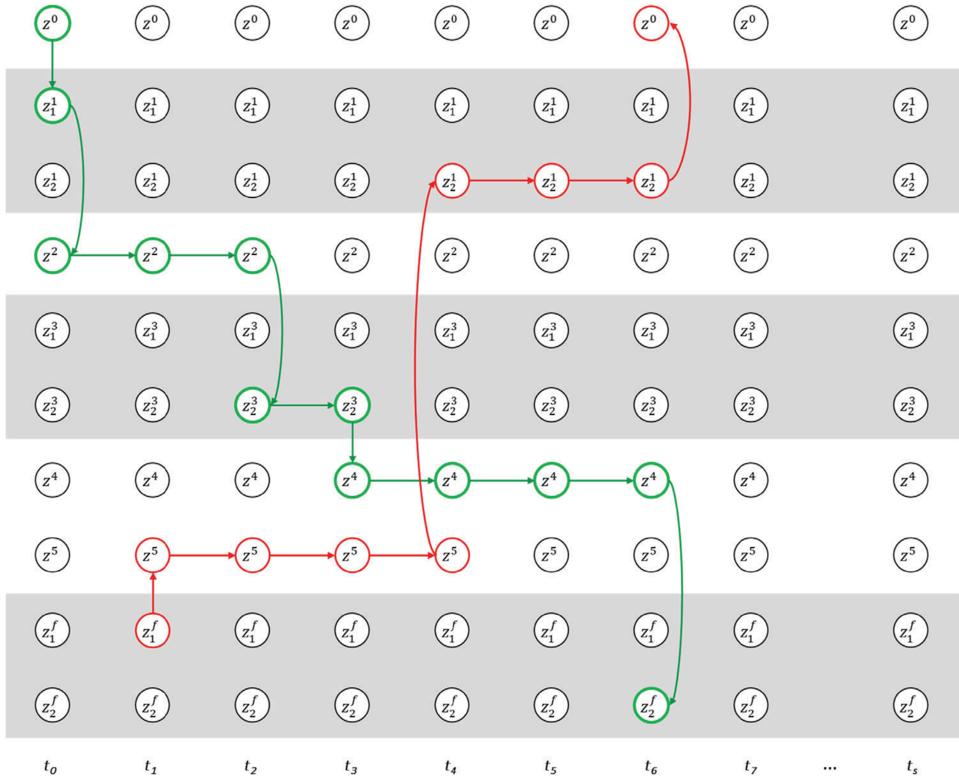

FIGURE 6 Example of paths on the operation-time-space-network

to the maritime terminal. These transfer operations are realized in different zones here called primary and secondary zones, respectively, the yellow and the orange in Figure 5. The waiting operations are realized in the shunting park.

Note that, it is necessary to know the track used for waiting operations in the rail station and in the shunting park, but it is not necessary for the execution of a transfer operation, since only one transfer operation for each type can be realized at each time; this means that a unique shunting operation can start only when the previous unique operation has been completed. The same is for the primary and secondary operations.

The limitations for the waiting operations are related to the number of tracks available in the rail station and the shunting park. Each track can receive one train. In some operative scenarios, it is possible to put more than one train on each track and there is a capacity limitation that is related to the length of the tracks. Note that also this situation can be modeled thanks to the proposed network.

Another capacity constraint is due to the limited number of teams for realizing the shunting operations in the whole port area.

Some time constraints are imposed in the different zones. In particular, the rail station imposes a fixed arrival (departure) time for each export (import) train, having to meet the scheduled activities for trains in the railway network. Terminals, having scheduled activities for ships, may impose a time window for the arrival (departure) of export (import) trains. Some time constraints can be imposed also for activities to perform in the different zones, for instance, a train might have to start (finish) an activity within a deadline.

Given a time horizon and the schedules of the trains on the railway network, on one side, and the time windows for entering and leaving the maritime terminals, on the other side, the problem consists in the scheduling of the operations for transferring the freight trains within the port area, in order to minimize the total time spent by the trains in waiting operations, while satisfying all capacity and time constraints.

All these operations can be represented on the operation-time-space network, and we can solve the scheduling of these operations for the import and export trains by solving a network flow model.

Figure 6 shows the operation-time-space network used for modeling a port area, that is, a physical network composed by the rail station with two tracks, one shunting park with two tracks, as well and two terminals. The paths of one export train and one import train that have to be transferred within this area are depicted. In particular, the two paths represent the sequence of operations performed by each train: the green for the export train and the red for the import one.

Let us suppose that the import train (red path) must leave the terminal z_1^f within the time window $[t_0; t_1]$, and its departure time from the rail station is t_6. The export train has the arrival time at the railway station t_0 and has to be at destination, that is, inside the terminal z_2^f within the time window $[t_6; t_8]$.

Looking at Figure 6, the export train (the green one) arrives from the rail network (z^0) in t_0 at the first track of the station (z_1^1) and it starts immediately the shunting operation through primary zone (z^2). The train performs this operation for two time intervals (i.e., $[t_0;t_1)$ and $[t_1;t_2)$). In time instant t_2 it finishes its primary operation and starts waiting operation in the second track of the shunting park (z_2^3). The train waits for one time interval and then, in time instant t_3 starts the secondary operation in z^4. It performs this latter operation for three time intervals until t_6. Finally, in t_6, once finished its secondary operation, the train arrives at its destination (terminal z_2^f).

As far as the import train is considered (the red path), it is ready at origin, that is, terminal z_1^f, in time instant t_1. In this time instant it starts the shunting operation through unique zone (z^5). It performs this operation for three time intervals until t_4; in t_4 it starts to wait in the second track of the train station (z_2^1). The train waits for two time intervals and then, in time instant t_6, departs on the railway network (z^0). Suppose to have two shunting teams in each t, the depicted example shows a simultaneous use of these two teams in t_1, t_2, and t_3.

Summarizing, the network proposed in Section 2 is useful to solve the scheduling of port shunting operations, that is an emerging problem in rail-sea terminals. Presently, we consider only the transfer of whole trains. Anyway, the proposed network can be helpful also for scheduling the operations for disassembled portion of trains, that is, their transfer inside the ports area and their reassembled for the departure on the railway network (for more details on this problem interested readers can refer to [16]). The case study here discussed is related to a port area in the North of Italy, having a layout as that described above and depicted in Figure 5. In particular, there is a port system with a rail station, a shunting park, and four terminals; the rail station and the shunting park have 10 tracks each. The shunting manager usually plans the activities for 1 week at a time, thus we consider a planning horizon T of 6 days, discretized into time intervals of 10 min.

In the next subsections, first we briefly introduce the operation-time-space network used for modeling this problem, and then we report some preliminary results related to both a simple example with a limited number of trains in a shorter planning horizon, and random generated instances.

3.1 | The operation-time-space network for scheduling port shunting operations

Let us introduce the basic elements to define the operation-time-space network to model the case study under investigation, based on the layout shown in Figure 5:

$O = \{z^1, z^2, z^3, z^4, z^5, z^0, z^f\}$ the set of activities (operations) to perform, included the arrival to/the departure from the railway network and terminals;

$O^+ = \{z^1, z^3\}$ the set of activities to perform on different tracks, that is, waiting operations at the rail station and at the shunting park;

$O^- = \{z^2, z^4, z^5\}$ the set of activities (operations) to perform without distinguishing the resources;

O_j the set of operations that must be performed by train j, $\forall j \in J$, where J is the set of trains given by the union of import trains (i.e., I) and the export trains (i.e., E).

To define the vertical arcs of the network, as explained in Section 2, we need to know the set of couples of compatible activities. For dealing with import and export trains we need to distinguish $O^C(E)$ the set of couples of operations that can be executed in sequence on an export train, and $O^C(I)$ the set of couples of operations that can be executed in sequence on an import train.

The generated vertical arcs related to an export train (i.e., related to the set $O^C(E)$) are shown in Figure 7.

Note that, thanks to the definition of vertical arcs based on the concept of compatible activities, we are able, when constructing the network, to reduce the network size, in particular the number of arcs. Moreover, we adopt another strategy for reducing the flow model size; we define the decision variables only if "time consistent" with respect to time constraints.

For each export train, both the arrival time at the rail station and the time window for entering the maritime terminal of destination are known, and for each import train, both the departure time from the rail station and the time window for exiting the maritime terminal of origin are known too. Thus, it is possible to assume that the shunting operations on each train have to be performed within a specific slot of the whole time horizon considered, here called competence slot. For export trains, the competence slot starts when the train physically arrives outside the port area and ends in accordance with the upper bound of its time window for entering in its terminal of destination. For import trains, the competence slot starts in accordance with the lower bound of the time window of the terminal of origin and ends when the train has to depart from the rail station. Thus, for example, let us consider an export train j, variables x_{aj} are generated in accordance with both $O^C(E)$ and the time constraints of train j.

The proposed network and the related flow formulation have been used to solve real problems related to some port areas in the North Italy. The results of one case are reported here below.

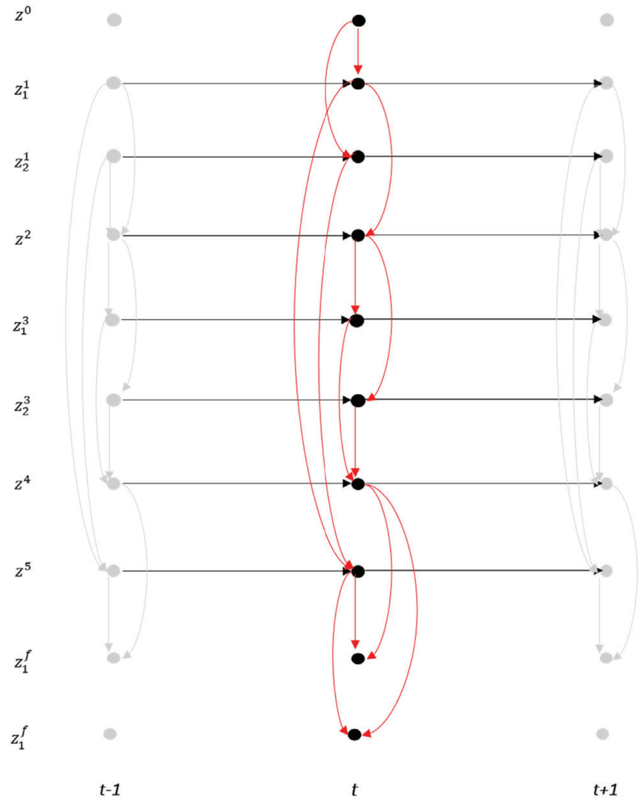

FIGURE 7 Vertical arcs for export trains in a time section of the network

Train Id	e /s	E/I	Maritime Terminal	e^{min}/s^{min}	e^{max}/s^{max}
1	18:00:00	E	1	19:00:00	22:00:00
2	15:27:00	E	1	17:00:00	19:00:00
3	07:20:00	I	2	04:00:00	07:00:00
4	11:57:00	I	1	09:00:00	10:30:00
5	23:00:00	I	2	18:00:00	21:00:00
6	13:38:00	I	1	08:00:00	10:00:00
7	22:21:00	I	1	20:00:00	21:20:00
8	14:32:00	I	2	10:00:00	11:00:00
9	00:54:00	I	2	23:00:00	23:50:00
10	14:48:00	E	1	16:00:00	17:00:00

FIGURE 8 Trains input data

3.2 | A simple example

In this section we describe in detail a simple case, stressing the time and the capacity constraints required for formulating this scheduling problem. Note that, the simple example is derived from the real case, by considering a shorter time horizon and a limited number of trains.

Ten trains must be managed in a day, that is, on Monday, and the main aim is to minimize the total waiting time spent by the trains in the tracks of both the station and the shunting park. The trains data are shown in Figure 8, which reports for each train the train Id, the arrival/departure time (s_j and e_j), the train cycle (i.e., the export/import cycle), the terminal of origin/destination, and the time window in which it has to arrive at destination $[e_j^{\min}, e_j^{\max}]$/depart from origin $[s_j^{\min}, s_j^{\max}]$.

All trains have to pass through the primary and the secondary zones and in the shunting park (i.e., $O_j = \{z^1, z^2, z^3, z^4, z^0, z^f\}$). The time that every train has to spend for executing operations in primary zone and secondary zone are known and equal to 20 and 60 min, respectively.

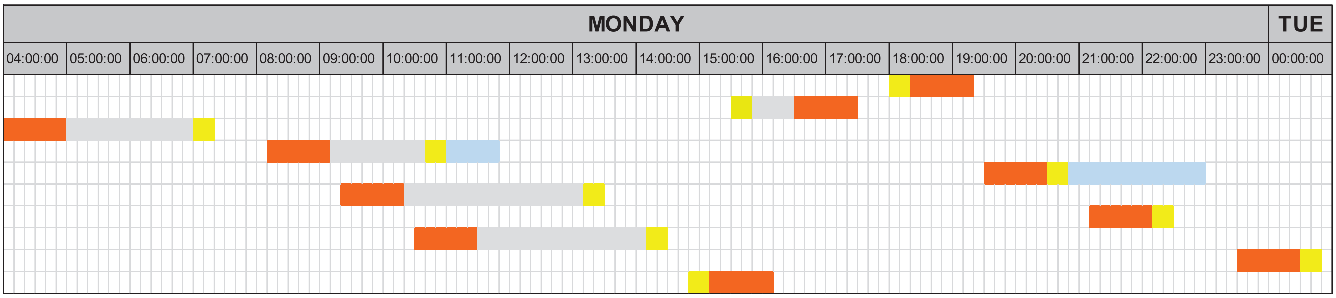

FIGURE 9 Example: Graphical representation of the solution

In each t there are the following capacity limitations:

(1) no more than one operation in each of the following zones can be executed: z^2, z^4, z^5, that is, only one primary operation, only one secondary operation, and only one unique operation;
(2) no more than two operations can be simultaneously executed in the following zones: z^2, z^4, z^5, since there are two shunting teams;
(3) no more than one train can enter/leave each terminal, that is, z_1^f, z_2^f;
(4) no more than one train can wait in each track of the shunting park, that is, in z_1^3, z_2^3.

We can model limitations (1) and (2) thanks to constraints CC1 and CC3, while we model limitations (3) and (4) by constraints CC2.

As far as time limitations are considered, we have to distinguish between export and import trains:

- each export train j arrives at the rail station in its due time (s_j), and:

in s_j it has to start either its shunting operation (in this example a primary operation) or its waiting in a track of the station; TC3 are used.

in $[e_j^{\min}, e_j^{\max}]$ it has to end its shunting operation (in this example a secondary operation) and to enter its destination terminal; constraints TC1′ are used.

- each import train j:

in $[s_j^{\min}, s_j^{\max}]$ has to leave its origin terminal and start its shunting operation (in this example a secondary operation); TC1 are used.

in e_j it has to end either its shunting operation (in this example a primary operation) or its waiting in a track of the station, for its departure on the railway network; constraints TC3′ are used.

The schedule of the operations of the considered trains has been defined by solving the flow model based on the operation-time-space network with the commercial solver Gurobi 8.1.0. The model has 3774 variables and 2346 constraints, and it has been solved in few seconds.

Figure 9 is a graphical representation of the obtained solution. The schedule of the operations that the 10 trains listed in Figure 8 have to perform are shown using different colors: yellow and orange represent shunting operations in primary and secondary area, respectively; light blue and gray represent the waiting operations on the tracks of the rail station and the shunting park.

Let us explain the schedule for the export train number 2 and the import train number 7, in order to clarify the example. Looking Figure 8, the export train number 2 arrives at 15:27:00 on a track of the train station and has to reach terminal 1 between 17:00:00 and 19:00:00 (the time window in which the terminal is ready to receive the train). From Figure 9 we can note that it starts the operation through primary zone at 15:30:00 and ends it at 15:50:00, while it starts the operation through secondary zone at 16:30:00 and ends it at 17:30:00 (respecting its time window). In this case, when it arrives at the rail station (15:27:00) has to start almost immediately the first operation, without waiting time on a track of the station, while it has to wait 40 min (i.e., the difference between the initial time of the second operation and the ending time of the previous) in the park.

The import train 7 has to depart from the rail station at 22:21:00, leaving terminal 1 between 20:00:00 and 21:20:00. It has to perform the operation in the secondary zone before the one in the primary zone.

It starts the operation through secondary zone at 21:00:00 (respecting its time window) and ends it at 22:00:00. Then, it performs the shunting operation through primary zone from 22:00:00 to 22:20:00, respecting its time path for departing (22:21:00). Note that, train 7 has not to wait neither in the shunting park nor in the rail station before departing.

The total waiting time of the optimal solution depicted in Figure 9 is 760 min, 180 min spent in the station and the remaining in the park.

# Trains	Arrival/Departure distribution		Wait Station h/train	Wait Park h/train	Total wait	# Variables	# Constraints	CPU Time
30	Homogeneous	2 days	10.72	4.10	14.82	121234	62581	22.5
	Homogeneous	1 day	4.07	0.80	4.86	73363	44929	17.0
	Homogeneous	3 shifts	1.52	1.08	2.61	42005	34035	2.6
	Compact	2 days	7.79	1.37	9.16	130113	66543	25.1
50	Homogeneous	2 days	9.74	8.71	18.45	232522	104555	207.1
	Homogeneous	1 day	4.28	0.89	5.17	136119	68340	58.7
	Homogeneous	3 shifts	1.71	0.82	2.53	71296	45250	3.4
	Compact	2 days	6.44	3.01	9.45	236989	107969	746.0

FIGURE 10 Results comparison with respect to arrival/departure distributions

# Trains	Time windows characteristics	Wait Station h/train	Wait Park h/train	Total wait
30 (Distr. Homo 1 day)	Fixed 1 hour	4.64	0.96	5.60
	Fixed 6 hours	3.24	1.34	4.59
50 (Distr. Homo 1 day)	Fixed 1 hour	5.31	0.55	5.86
	Fixed 6 hours	3.15	1.26	4.42

FIGURE 11 Results comparison with respect to the time windows

Note that, the graphical representation of the solution is very helpful to visualize and check the use of the resources; thanks to it, it is possible to understand both critical points and the available changes to perform.

3.3 | Computational results on random generated instances

In this subsection, we show the results obtained by using the network flow model for solving random generated instances representing possible scenarios of the real case; these instances are characterized by 30 and 50 trains equally distributed between import and export in a time horizon of 6 days.

For each train the set of operations to perform is randomly generated, as well as its origin/destination terminal among the four terminals of the port area.

As far as the distribution of the trains in T is considered, four different sets of instances were generated, for testing the impact of the balanced and unbalanced arrivals and departures of trains.

In particular, the distribution can be either homogeneous or compact. Homogeneous means that the number of trains is constant for specific time intervals within the time horizon. We considered the following time intervals: i) 2 days (i.e., three time intervals of 2 days in the 6-day time horizon); ii) 1 day (i.e., six time intervals of 1 day in the 6 day time horizon); iii) a working shift (i.e., three shifts per day, thus 18 time intervals in the 6-day time horizon).

In the compact distribution, we use the 2-day time intervals and, in contrast with the homogeneous distribution, we impose that the 50% of the total number of trains arrive within the first interval, the 25% in the second and the third. Note that, respecting the distributions here explained, the arrival and departure time for each train within the time intervals is randomly assigned.

Going ahead to the arrival and departure times, the time windows in which each train can enter/leave the terminal are randomly assigned; moreover, the time windows width can be either 1 or 6 h .

Note that, an interval of 3 h between the arrival/departure time and the time windows is always imposed in order to be sure that the times randomly generated grant at least the required time to perform the operations.

The flow model based on the proposed operation-time-space network has been implemented in Python 3.7, and solved by commercial solver Gurobi 8.1.0, on a machine intel (R), i5, 7200U CPU, 2.5 GHz, 8.00 GB RAM.

In Figures 10 and 11 the obtained results are shown. In particular, Figure 10 permits to compare the results in terms of the total waiting time spent by each train when different arrival/departure distribution is considered. As expected, having a better distribution of trains during the whole time horizon T (here represented by homogeneous generation in each shift of 8 h) permits to reduce the total waiting time.

The last columns are related to the size (number of variables, number of constraints) of the solved model and to the computational time. The reader can note the there is a high variability in the number of variables and constraints that ranges from 42 005 to 130 113 variables, from 34 035 to 66 543 constraints. This high difference in the model size is strongly related to the characteristics of the generated instances and even more notable when reducing variables in accordance with the competence slot, that are generally smaller when working with shifts.

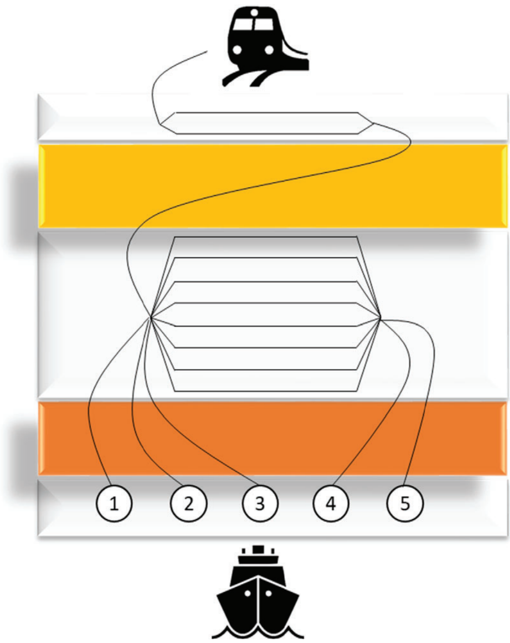

FIGURE 12 Another rail-sea modality exchange node configuration

The average time required for solving instances with 30 trains is 16.8 s, while for the instances with 50 trains is 253.8 s.

Focusing on the homogeneous distribution of the train with respect to the days of the planning horizon, Figure 11 permits to understand that larger time windows permit to better organize the activities in the shunting zone. In fact, the larger is the time window, the lowest is the total waiting time spent by each train.

Generally, it is possible to note that the major part of the waiting time is spent by trains at the rail station. Unfortunately, the trains schedule on the railway network are fixed and represent for the shunting manager strong constraints to respect.

3.4 | Another solved port shunting operation scheduling problem

Analyzing the layout of the ports areas in Italy, where the scheduling of rail shunting operations must be managed, we noted that they present different layouts, different capacity, and time constraints. We are able to use the proposed operation-time-space-network to model and solve their scheduling problems regarding rail shunting operations.

In particular, we have used the proposed network to face the shunting scheduling problem for a port area having a configuration as the one depicted in Figure 12, in which the shunting manager deals with both whole trains and portions of them (i.e., when it is required to split a train into two or more portions before transferring it inside its maritime terminal of destination).

As already said, Rusca et al. [16] analyze this case, but the authors assume that each portion has a different terminal as destination; our case is different, because in Italy, for the lack of space and for the length of tracks inside the terminals, the portions of a train have often the same destination.

Relating to the layout showed in Figure 12, we are able to schedule a number of trains that ranges from 20 to 50 with a CPU time ranging from 10 min to 2 h. We noted that the CPU time is strongly related to both the complexity of the network and the real distribution of import and export trains; it become higher when increasing the shared available resources. In fact, when some resources are shared among more terminals, the difficulty in solving the problem increases (i.e., the problem with only one terminal is solved in 1 min).

4 | OTHER USES OF THE PROPOSED NETWORK AND CONCLUSIONS

This article aims at describing an innovative time-space network that can be easily used to face different problems in which time and space aspects are crucial.

Beside the problems described in the previous section, some works using the proposed network are in progress. In particular, we are formulating a network design problem on the proposed network to define the optimal layout of a port area together with the required resources necessary to manage a given number of trains in a certain planning horizon.

Another problem that we are facing by using the proposed network is a timetable problem for a school, that in accordance with the new regulations and COVID restrictions has to satisfy different capacity and time constraints.

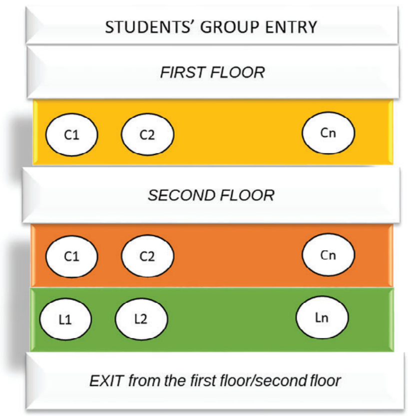

FIGURE 13 School layout and activities

The school layout is shown in Figure 13; we have to manage different groups of students that have to be assigned to some classrooms and to some labs, in such a way to satisfy some capacity constraints related to the classrooms, labs, floors, and entry/exit. Moreover, we have some time constraints for the entry and the exit of the students, and for the duration of the classrooms and labs.

Just to complete the analysis, in the next subsection we will report a well-known scheduling problem, the quay crane scheduling problem (QCSP), in order to present how to model it by the operation-time-space network and in order to evaluate the performances of the related flow model.

4.1 | Modeling and solving the QCSP with the operation-time-space network

An important problem faced in port areas, inside the maritime terminals, is the quay crane scheduling problem (QCSP). QCSP is the problem of scheduling the loading and unloading operations of a vessel performed by a set of cranes. The main aim is to minimize the total vessel handling time, while satisfying some constraints generally related to the movement/operation of cranes and to relation among tasks [14].

Presently, our main aim is to show how the operation-time-space-network proposed in Section 2 can be used for modeling and solving real quay crane scheduling problems, also evaluating the performances of the network flow model for solving QCSP. For this specific purpose we will refer to the benchmark suite proposed in [7].

Let us suppose to have a set of QCs that have to execute some tasks in the bays of a vessel. The time horizon is equal to the service time that the terminal has to grant to the ship. The operation-time-space network described in Section 2 can be adapted to model QCSP. We use the nodes of the network for representing the bays of the ship, the buffers that here permit the movement of the cranes between two adjacent bays, the entry/the exit of the QCs in/from the network, that is, representing the starting and the ending of each QCs work.

We use the arcs for representing both the movement of the QCs among the bays and the operations realized by each QC in the bays of the ship. We distinguish horizontal and vertical arcs as shown in Figure 14. In this figure a simple example with four bays (and four movement nodes) is depicted, and the focus is on a specific t. Vertical arcs connect: i) the entering node b^0 to all the bays of the vessel; ii) each bay node to the final node b^f; iii) a bay b^i to its buffer m^i (movement node); iv) a movement node m^i to the two adjacent bays of b^i (i.e., b^{i-1}, b^{i+1}). The horizontal arcs between bay nodes and between movement nodes represent the execution of tasks in the bays and the movements of the QCs (in $[t, t+1)$).

For what concerns the tasks that we have to schedule, we refer to *complete bays* (see [14] for more details). Thus, we suppose to know the total work that a crane has to execute in a bay. If this is not the case, we can determine the sequence of tasks that must be performed within a bay of the ship and considering all these tasks as one. Moreover, the time required to complete the task in each bay is given. Note that, differently from the papers dealing with complete bays, we permit to share the work of a bay between two cranes; in this way, we avoid the problem stressed in [14] of reducing the achievable service quality. Moreover, the starting position of each QC is given.

The problem consists in determining a sequence of bays that each QC has to operate, in such a way to minimize the total handling time, complete the tasks, avoid crossing between cranes, respect safety rules (i.e., distance required between two QCs), eventual time constraints related to both QC working time and bay working time and, finally, to respect capacity constraints.

For what concerns the time constraints, thanks to the operation-time-space network, we are able to impose an initial and a final operation time for each QC, if different from the beginning and the final t in the considered time horizon T. We are also able to impose to work on a bay within a deadline, after a given time and during a fixed time window. Note that, these constraints

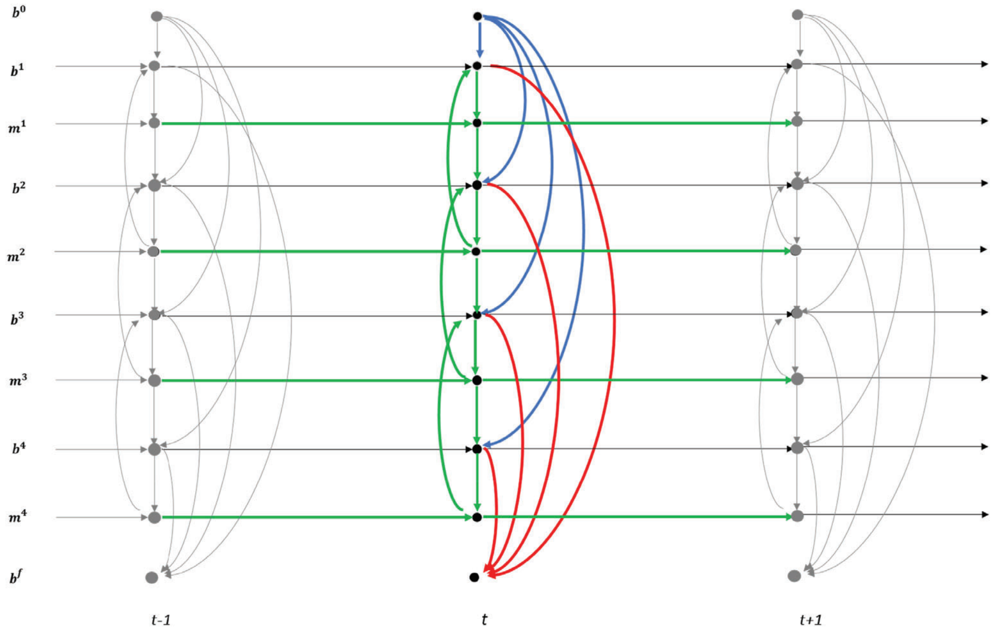

FIGURE 14 Vertical and horizontal arcs in the quay crane scheduling problem network

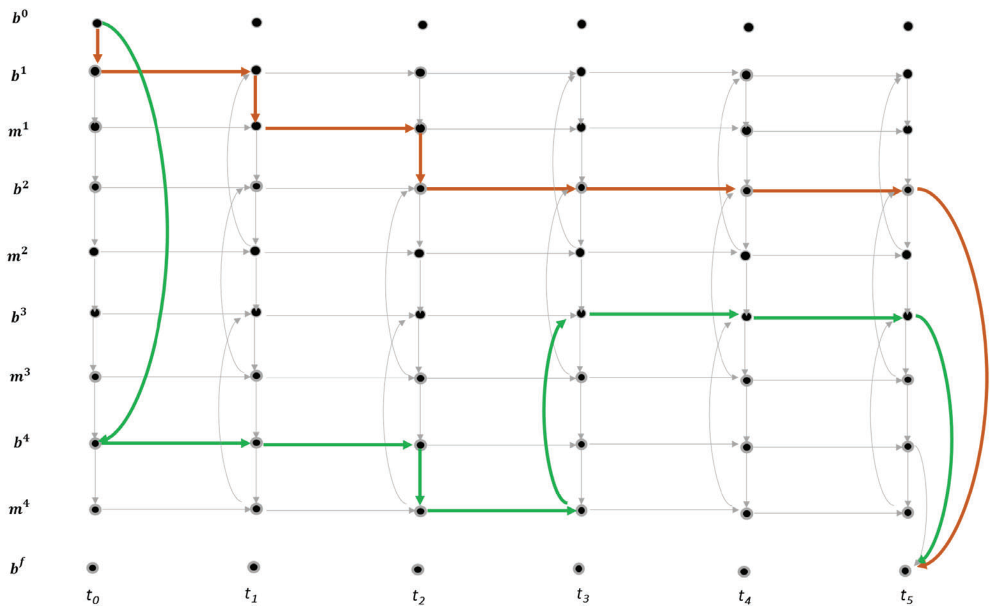

FIGURE 15 Paths on the operation-time-space network of two cranes operating four bays of a vessel

can be related to some operational requirements due to the yard management and necessary to grant efficiency during pick up operations in the yard.

For what concerns the capacity constraints, in each t, the number of QCs working the ship must respect the maximum number of available QCs and each bay can be worked by at most one QC.

An example of the obtained solution through the flow model based on the operation-time-space network, adopted as explained above to model QCSP, is reported in Figure 15.

Two QCs start their tasks in t^0, respectively in bay b^1 and b^4. QC1 (the orange one in the figure), after having completed the task in b^1, moves to b^2 (passing through m^2) and in t_2 starts to work in b^2. In t_5 it finishes and leaves the network. QC2 (the green one in the figure) after having completed (in t_2) the task in b^4, moves to b^3, operates b^3 until t_5; in t_5 it leaves the network. Note that, in this simple example we have not included the safety distances between QCs.

The QCSP is a well-studied problem for which a lot of test benchmark instances have been generated, see, for example, [7] and [14]. We test our flow model based on the operation-time-space network developed for the QCSP, using the benchmark suite proposed by Kim and Park [7]. In particular, the number of tasks range from 10 to 25 and two or three QCs are considered. The results, obtained by using Gurobi 8.1.0, have been compared with those shown in [14]. We are able to solve the smaller

Instance	# bays	# QCs	# variables	# constraints	CPU time (secs)
1	10	2	8120	3673	2
2	10	2	14376	7333	99
3	10	2	28776	14653	385
4	15	2	11120	5478	155
5	15	2	22280	10938	868
6	15	2	44600	21858	-
7	20	2	14946	7283	180
8	20	2	29946	14543	3059
9	20	2	59946	29063	-
10	20	3	20708	10885	55
11	20	3	41468	21745	3543
12	20	3	82988	43465	-
13	25	2	18300	9088	158
14	25	2	36660	18148	908
15	25	2	73380	36268	-
16	25	3	26270	13590	1608
17	25	3	52610	27150	-
18	25	3	105290	54270	-

FIGURE 16 Quay crane scheduling problem results

instances in a CPU time that ranges from 2 s to 6 min and 25 s, depending on the length of the planning horizon that is strongly related to the processing time of the tasks to execute. Figure 16 shows the obtained results, that are similar to those presented in [14] in case of instances with 20 bays and three QCs. For what concerns larger instances with 25 bays, we are able to solve three of six within 30 min (only one with three QC), while none has been solved by CPLEX in [14].

The number of available QCs impacts on the CPU time. We test the model also including some time constraints that are not included in the test of [14]. These constraints have not a great impact on the computational time. The more relevant aspect for the CPU time is the time horizon, and in particular, the number of intervals in which is split the time horizon. This impacts on the number of variables and constraints and, finally, on the CPU time. Thus confirming that it is very important to analyze the characteristics of the problem in order to reduce the network and the model size.

4.2 | Conclusions

The usage of the operation-time-space network seems to be really promising. Many real problems concerning the organization of activities in the space and along a time horizon can be modeled with the proposed network, and the associated flow formulations are generally effective to solve them. In particular, this network permits to include in the model a lot of different types of constraints that generally require more complex formulations (e.g., constraints for avoiding crossing between cranes). The results show that the capability of the operation-time-space networks (and their related flow models) may depend on the possibility of reducing the networks and the model size. The time is probably the most important factor acting on the complexity of the model. Summarizing, each problem that can be modeled with an operation-time-space network has specific characteristics; studying them it is possible to define strategies for reducing the network size and/or the model size and thus the complexity of the solution process.

DATA AVAILABILITY STATEMENT

Data are available on request due to privacy/ethical restrictions

REFERENCES

[1] R. Agarwal and Ö. Ergun, *Ship scheduling and network design for cargo routing in liner shipping*, Transp. Sci. **42**(2) (2008), 175–196.
[2] D. Ambrosino and V. Asta, *Intermodality and rail transport: Focus on port rail shunting operations*, in *Advances in Optimization and Decision Science for Society, Services and Enterprises*, Springer, New York, NY, 2019, 351–366.
[3] D. Ambrosino and L. Peirano, *Truck arrival management at maritime container terminals*, ECMS **2016** (2016), 114–120.
[4] C. Archetti and L. Peirano, *Air intermodal freight transportation: The freight forwarder service problem*, Omega **94** (2020), 102040.
[5] Y. Guan, K. H. Yang, and Z. Zhou, *The crane scheduling problem: Models and solution approaches*, Ann. Oper. Res. **203**(1) (2013), 119–139.
[6] F. Guerriero and L. Talarico, *A solution approach to find the critical path in a time-constrained activity network*, Comput. Oper. Res. **37**(9) (2010), 1557–1569.
[7] K. H. Kim and Y. M. Park, *A crane scheduling method for port container terminals*, Eur. J. Oper. Res. **156** (2004), 752–768.

[8] P. Li, P. Mirchandani, and X. Zhou, *Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork*, Transp. Res. B Methodol. **81** (2015), 103–130.
[9] J. Liu and X. Zhou, *Capacitated transit service network design with boundedly rational agents*, Transp. Res. B Methodol. **93** (2016), 225–250.
[10] J. Liu and X. Zhou, *Observability quantification of public transportation systems with heterogeneous data sources: An information-space projection approach based on discretized space-time network flow models*, Transp. Res. B Methodol. **128** (2019), 302–323.
[11] C. C. Lu, *Robust multi-period fleet allocation models for bike-sharing systems*, Netw. Spat. Econ. **16**(1) (2016), 61–82.
[12] G. Lu, X. Zhou, M. Mahmoudi, T. Shi, and Q. Peng, *Optimizing resource recharging location-routing plans: A resource-space-time network modeling framework for railway locomotive refueling applications*, Comput. Ind. Eng. **127**(2) (2019), 1241–1258.
[13] M. Mahmoudi and X. Zhou, *Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations*, Transp. Res. B Methodol. **89** (2016), 19–42.
[14] F. Meisel and C. Bierwirth, *A unified approach for the evaluation of quay crane scheduling models and algorithms*, Comput. Operat. Res. **38** (2011), 683–693.
[15] K. Murakami, *Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system*, Comput. Ind. Eng. **141** (2020), 106270.
[16] F. Rusca, M. Popa, E. Rosca, A. Rusca, M. Rosca, and O. Dinu, *Assessing the transit capacity of port shunting yards through discrete simulation*, Transp. Probl. Int. Sci. J. **14**(4) (2019), 101–112.
[17] L. Tong, X. Zhou, and H. J. Miller, *Transportation network design for maximizing space–time accessibility*, Transp. Res. B Methodol. **81** (2015), 555–576.
[18] L. Yang, Z. Di, M. M. Dessouky, Z. Gao, and J. Shi, *Collaborative optimization of last-train timetables with accessibility: A space-time network design based approach*, Transp. Res. C Emerg. Technol. **114** (2020), 572–597.
[19] Y. Yuan and L. Tang, *Novel time-space network flow formulation and approximate dynamic programming approach for the crane scheduling in a coil warehouse*, Eur. J. Oper. Res. **262**(2) (2017), 424–437.
[20] E. Zehendner and D. Feillet, *Benefits of a truck appointment system on the service quality of inland transport modes at a multimodal container terminal*, Eur. J. Oper. Res. **235**(2) (2014), 461–469.

How to cite this article: Ambrosino D, Asta V. An innovative operation-time-space network for solving different logistic problems with capacity and time constraints. *Networks.* 2021;78:350–367. https://doi.org/10.1002/net.22042