Network scheduling problem with cross-docking and loading constraints

Pedro B. Castellucci*[1], Alysson M. Costa†[2], and Franklina Toledo‡[3]

[1]*Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Brazil*
[2]*School of Mathematics and Statistics, The University of Melbourne, Australia*
[3]*Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Brazil*

Abstract

Cross-docking is a logistic strategy which can increase rates of consolidation, and reduce distribution and storage costs and delivery times. The optimization literature on cross-docking has mostly focused on the modeling and solution of problems considering a single cross-docking facility. Networks with multiple cross-docks remain rather unexplored and the few papers that deal with the problem do this by simplifying the geometry of the goods. We a mixed-integer linear programming model for optimizing distribution and delay costs for the transportation of goods in open networks with multiple cross-docks considering the three-dimensional aspects. We propose a logic-based Benders decomposition strategy which allow for the solution of larger instances when compared with those that can by handle by a . Experiments showed

Keywords: vehicle routing, three-dimensional packing, Benders decomposition, city logistics.

*Corresponding author: pedro.castellucci@ufsc.br
†alysson.costa@unimelb.edu.au
‡fran@icmc.usp.br

1 Introduction

In a logistic distribution system, two of the most expensive tasks are storage and order picking (Belle et al., 2012). eliminate both by synchronizing the input and the output flow of goods. In a cross-dock, goods are unloaded at an input dock, sorted according to their destination and loaded into trucks at an output dock, reducing storage requirements. This strategy aims at obtaining higher rates of consolidation, shorter delivery and lead times, and has been implemented successfully in many situations (Cook et al., 2005; Forger, 1995; Kinnear, 2006; Napolitano, 2011; Stalk et al., 1992; Witt, 1998). Figure 1 shows a schematic representation of a typical cross-dock.

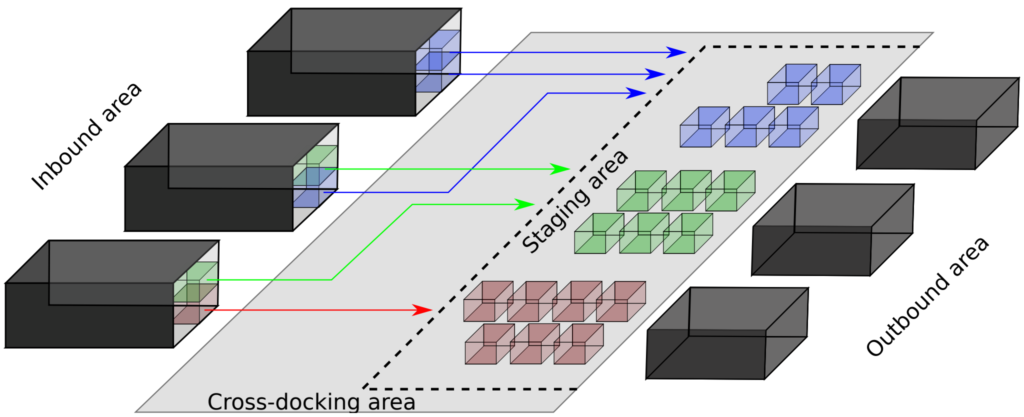


Figure 1: Schematic representation of a cross-dock.

The optimization literature on cross-docking has mostly focused on the operation of a single facility and scarcer are the papers dealing with networks (open or closed) with cross-docks (see Subsection 2.1). of incorporating two or three-dimensional aspects of the cargo (see Subsection 2.2).

To the best of our knowledge, this is the first time . More specifically, our contribution is two-fold. First, we define and model a distribution problem in a network with suppliers, consumers and consolidation facilities (cross-docks). Each consumer demands for boxes (with length, width and height) . Second, we proposed a reformulation of this model, which can be solved using the logic-based Benders framework (Hooker and Ottosson, 2003).. This separation allows for variations of the loading aspects to be considered within the same framework.

(i)

(ii)

The remainder of the manuscript presents a literature review (Section 2) on cross-dockingvehicle routing problems with loading constraints . Then, Section 3 formally defines the *network scheduling problem with cross-docking and loading constraints* and propose a model for the problem. a reformulation of the model and the solution procedure which makes use of the logic-based Benders framework. Results evaluating the model and the decomposition strategies are presented in Section 5, followed by the conclusions, in Section 6.

2 Literature review

Due to the lack of references on network distribution problems that consider packing/loading constraints not limited to the one-dimensional case, we present a literature review on cross-dock optimization and on vehicle routing with loading constraints. The cross-docking review (Subsection 2.1) shows current trends focus on optimizing the operation of a single facility rather than a network with consolidation centers. Furthermore, the few papers which considered networks with cross-dock simplify the capacity of the trucks and the demand of the consumer to the one-dimensional case. The importance of considering a more detailed geometric description of the goods is highlighted by the vehicle routing literature (Subsection 2.2).

2.1 Cross-docking review

The optimization literature in cross-docking has mostly focused on optimizing the operation of a single facility. Perhaps, the most popular optimization problem in this context is the scheduling of trucks at input and output docks. Some research on the case with one input and one output dock . Although the one-dock case is mostly unrealistic, authors agree that it might provide insights into promising solution methodologies and a general understanding of the behavior of larger systems. Multi-dock cases have recently been studied (e.g.). Different cross-dock characteristics were also modeled. Some consider time windows for trucks to be loaded or unloaded , limited resources for moving goods from inbound to outbound docks , goods' handling time/effort , traveling distance inside the cross-dock and limited internal storage . For detailed reviews, the reader is referred to the papers by Agustina et al. (2010), Stephan and Boysen (2011), Belle et al. (2012), Buijs et al. (2014), Guastaroba et al. (2016)and Gelareh et al. (2020).

The literature approaching transportation networks with cross-docking scarce. Most of the research exploring this kind of network focused on closed networks, i.e., the routes of the vehicles start and end at the same node . The open network scheduling problem with cross-docking remains, to the best of our knowledge, a rather unexplored case. Geoffrion and Graves (1974) were one of the firsts to study an open network setting with distribution centers similar to cross-docks. In their seminal work, the authors formulated a mixed-integer linear programming model for the selection of distribution centers to be used as intermediate facilities for suppliers and consumers. For the solution of the model, a Benders decomposition proved its efficacy in finding optimal solutions. This is one of the first successful applications of Benders decomposition to a practical/realistic problem (Costa, 2005). Chen et al. (2006) proposed a model and a neighborhood search heuristic for the open case with multiple cross-docks. Yu et al. (2016) proposed the *Open Vehicle Routing Problem with Cross-docking*, which is related to the problem studied by Chen et al. (2006) but without time windows and with a single cross-dock. Moreover, Hernández et al. (2011) contemplated the case with multiple carriers which needed to move goods from origin to destination allowing collaboration among the carriers. The authors proposed a model for identifying a set of routes which minimize the total transportation costs.

Even though there is some research on open network problems , there are important differences related to the case here. Geoffrion and Graves (1974) approach a more tactical situation, deciding the flow of goods through consolidation centers and which centers to open. We focus on an operational problem, deciding on where to send trucks (rather than flows), how to recombine boxes into trucks from different suppliers. Chen et al. (2006) consider a network with two layers, similar to ours, but uses a discrete-time horizon to manage the cross-docks inventory levels, whilst we focus on the distribution aspects, limiting the capacity of trucks. Hernández et al. (2011) focus on a collaborative network, in which carriers may decide when and from which node they want to enter and exit this network. In our case, all participants of the network are previously defined and all supply and demand is always part of the same network. , Yu et al. (2016) consider a single cross-dock manager that outsources pickup and delivery activities. Multiple visits in pickup and delivery open routes are allowed and the synchronization at the cross-dock is achieved by forcing the fleet to arrive at the cross-dock at the same time. Alternatively, we consider multiple cross-docks, with single delivery visits and achieve synchronization by the availability of the outbound goods.

Additionally, we focus on the open network scheduling problem with multiple cross-docksdemand goods from different suppliers but instead of sending the goods directly, a supplier must ship them to a cross-dock where its load is consolidated with that of other suppliers. This strategy enables a full truckload to be sent to the consumers and is particularly interesting in city logistic contexts. In city logistics, the use of urban distribution centers (like cross-docks in the surroundings of the city), increase consolidation rates, reducing the number of vehicles used, and improving traffic in denser urban regions.

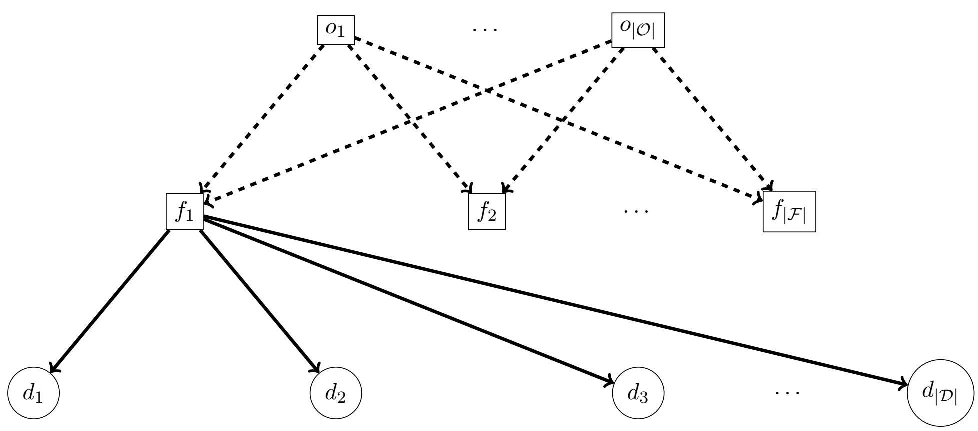


Figure 2: Scheme of a logistic network with cross-docks (consolidation centers).

2.2 Vehicle routing with loading constraints review

The importance of considering the geometry of vehicles and goods, particularly the three-dimensional geometry, is demonstrated by decades of research in container loading problems (Bortfeldt and Wäscher, 2013; Zhao et al., 2016). Approaching vehicle routing and loading decisions separately may overlook structural dependencies between the two problems (Vega-Mejía et al., 2019). Therefore, there has been a recent trend of incorporating these characteristics (2D or 3D) into vehicle routing problems (Pollaris et al., 2015). Thus, the decisions related to the vehicle routing problem with loading constraints are: (i) to position boxes inside the vehicles and (ii) assign delivery nodes to vehicles and the sequence of the visits. Even though there are exceptions, most of the literature focuses on the case with rectangular boxes and rectangular containers (trucks). Thereby, the demand from the consumers is given as a set of rectangular boxes (length, width and height, for the three-dimensional case). Also, the capacities of the trucks are given as a rectangular container (length, width and height).

The integration between the vehicle routing problem and the loading problem may be classified into different categories. on the capacitated vehicle routing problem with three-dimensional dimensional loading constraints (3L-CVRP) and some of its variations (e.g. the pickup and delivery problem). , we refer the reader to Pollaris et al. (2015). The 3L-CVRP is most frequently solved using a two-stage approach in which the routing decisions are done first and then a routine to solve the packing/loading problem is called to solve the loading problems. One of the exceptions is the paper by Reil et al. (2018). The authors proposed a packing-first routing-second approach in which the loading problem is solved using a tabu search and then an evolutionary algorithm minimizes the number of vehicles with a tabu search to minimize total travel time. This strategy is able to incorporate different variations of the routing problem, as stated by the authors.

The 3L-CVRP was introduced by Gendreau et al. (2006) motivated by a furniture transportation problem and the authors proposed a solution method based on tabu search. Since then, many heuristic approaches were applied to the 3L-CVRP and its variations. Zhu et al. (2012) also used a tabu search for the routing and proposed improvements on classical constructive heuristics for the loading problem. Bortfeldt and Yi (2020) proposed a local search for the routing and genetic algorithm for the loading for a variation which considers split-deliveries. Männel and Bortfeldt (2016) extended the pickup and delivery problem to incorporate loading decisions. The routing decisions were performed by a variable neighborhood search and the loading by a heuristic tree search. Koch et al. (2020)

used a tabu search for the routing and different heuristics for the loading for the vehicle routing with backhaul (i.e. goods have to be delivered from the depot and pickup from the customers). One of the few papers that consider the problem in a single stage is by Ceschia et al. (2013). The authors proposed a combination of simulated annealing with large neighborhood search to minimize total routing cost ensuring the feasibility of the loading.

Exact and model-based solution approaches are . Junqueira et al. (2013) proposed a mixed-integer linear programming model to minimize the total cost of the routes and consider some practical constraints for the loading (e.g. vertical stability and fragility of the boxes). Hokama et al. (2016) used techniques from branch-and-bound, constraint programming and meta-heuristics to design a branch-and-cut algorithm for the 3L-CVPR (and for the two-dimensional version). Massen et al. (2012) proposed a heuristic column generation procedure in which the columns (routes) have to satisfy unknown constraints (e.g. loading constraints). The columns are generated by a constructive heuristic guided by pheromones and the feasibility is checked by a separate procedure. Moura (2019) proposed a model based heuristic with a construction and local search phases for the vehicle routing with loading constraints.

Even though the integration of loading and network scheduling decisions is present in the vehicle routing literature (specially in closed networks), references considering such geometrical features in the cross-docking literature

2.3

cross-docks

2.4

is presented in Section 3.

3 Problem definition and model

We focus on distribution networks in which a set of suppliers (origins), $\mathcal{O}$, satisfies demands from a set of consumers (destinations), $\mathcal{D}$, by shipping their goods through intermediate (cross-docking) facilities, $\mathcal{F}$ (see Figure 2). Leaving each origin $o \in \mathcal{O}$, there is a set of shipments $k \in \mathcal{K}_{\mathcal{O}}$ and each shipment contains a set of boxes, or goods, $\mathcal{B}_k$. Let ℓ_{ia} be the length of each box $i \in \mathcal{B} = \bigcup_{k \in \mathcal{K}} \mathcal{B}_k$, along axis $a \in \mathcal{A} = \{\text{x-axis}, \text{y-axis}, \text{z-axis}\}$. Also, let UT_k and ST_k be the time required to unload and sort shipment $k \in \mathcal{K} = \bigcup_{o \in \mathcal{O}} \mathcal{K}_o$, respectively. Each cross-dock $f \in \mathcal{F}$ has a set of available vehicles, $\mathcal{W}_f$, each with length L_a along axis, $a \in \mathcal{A}$, and loading time LT_w, $w \in \mathcal{W} = \bigcup_{f \in \mathcal{F}} \mathcal{W}_f$. Each destination $d \in \mathcal{D}$ demands for the set of boxes $\mathcal{B}_d \subset \mathcal{B}$. All boxes demanded are supplied (i.e. $\bigcup_{k \in \mathcal{K}} \mathcal{B}_k = \bigcup_{d \in \mathcal{D}} \mathcal{B}_d$). Furthermore, let the cost and time of transporting goods from supplier $i \in \mathcal{O}$ to cross-dock $j \in \mathcal{F}$ or from cross-dock $i \in \mathcal{F}$ to consumer $j \in \mathcal{D}$ be c_{ij} and t_{ij}, respectively. Finally, there is a deadline τ_d associated with each consumer $d \in \mathcal{D}$. All trucks must reach their Also, we assume that each truck visits at most a single demand point (consumer), but a consumer might be visited by multiple trucks. Finally, let $\overline{u}_{if}$ be the time from which box $i \in \mathcal{B}$ is available for loading in a truck in facility $f \in \mathcal{F}$ (we show how these parameters can be computed later in this section). All parameters are summarized in Table 1.

The goal is to find an assignment of the shipments leaving suppliers to cross-docks and of boxes to trucks which minimizes total distribution and delay costs. For this, let Δ_{kf} be binary variables indicating whether shipment $k \in \mathcal{K}$ is assigned to cross-dock $f \in \mathcal{F}$ and p_{iw} be binary variables indicating whether box $i \in \mathcal{B}$ is assigned to truck $w \in \mathcal{W}$. Also, binary variables v_{wd} define whether truck $w \in \mathcal{W}$ is assigned to destination $d \in \mathcal{D}$. Moreover, there are continuous non-negative variables m_w the time a truck $w \in \mathcal{W}$ is ready for departure and e_{wd} the lateness of delivery truck $w \in \mathcal{W}$ for consumer $d \in \mathcal{D}$. To ensure the feasibility of truck loading, we also need to position each box inside the respective truck. For this, let x_{ia} be non negative variables indicating the position of box $i \in \mathcal{B}$ along axis $a \in \mathcal{A}$ and $r_{ija}^{af}(r_{ija}^{bf})$ be binary variables indicating whether box $i \in \mathcal{B}$ is after (before) box $j \in \mathcal{B}, j > i$, in axis $a \in \mathcal{A}$. In summary, we have the variables in Table 2. We formulate the resulting network scheduling problem with cross-dock (NSCD) (1)–(11).

$$\text{(NSCD)Min} \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} \sum_{f \in \mathcal{F}} c_{of} \Delta_{kf} + \sum_{f \in \mathcal{F}} \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} e_{wd} \tag{1}$$

subject to:

Table 1: Summary of parameters for the network scheduling problem .

$\mathcal{O}$:	Set of suppliers.
$\mathcal{D}$:	Set of destinations.
$\mathcal{F}$:	Set of cross-docks.
$\mathcal{K}_{\mathcal{O}}$:	Set of shipments leaving supplier $o \in \mathcal{O}$.
$\mathcal{K}$:	Set of all shipments $\left(\bigcup_{o\in\mathcal{O}} \mathcal{K}_o\right)$.
$\mathcal{B}_k$:	Set of boxes in shipment $k \in \mathcal{K}$.
$\mathcal{B}$:	Set of all boxes $\left(\bigcup_{k\in\mathcal{K}} \mathcal{B}_k\right)$.
ℓ_{ia}:	Length of box $i \in \mathcal{B}$ along axis $a \in \mathcal{A}$.
UT_k:	Unloading time of shipment $k \in \mathcal{K}$.
ST_k:	Sorting time of shipment $k \in \mathcal{K}$.
$\mathcal{W}_f$:	Set of trucks available at cross-dock $f \in \mathcal{F}$.
$\mathcal{W}$:	Set of all available trucks $\mathcal{W} = \bigcup_{f\in\mathcal{F}} \mathcal{W}_f$.
L_a:	Length of each truck along axis $a \in \mathcal{A}$.
LT_w:	Loading time of truck $w \in \mathcal{W}$.
c_{ij}:	Cost of moving from a supplier (cross-dock) i to a cross-dock (destination) j .
$\overline{u}_{if}$:	The time from which box $i \in \mathcal{B}$ is available for loading in a truck in facility $f \in \mathcal{F}$.
$\mathcal{B}_d$:	Set of boxes demanded by consumer $d \in \mathcal{D}$.
t_{ij}:	Travel time between points $i \in \mathcal{O} \cup \mathcal{F}$ and $j \in \mathcal{F} \cup \mathcal{D}$.
τ_d:	Deadline associated with client $d \in \mathcal{D}$.
R:	Penalty per unit of time delay .

Table 2: Summary of the variables for the network scheduling problem .

Δ_{kf}:	Binary variables indicating whether shipment $k \in \mathcal{K}$ is sent to facility $f \in \mathcal{F}$.
p_{iw}:	Binary variables indicating whether box $i \in \mathcal{B}$ is loaded into truck $w \in \mathcal{W}$.
v_{wd}:	Binary variables indicating whether vehicle $w \in \mathcal{W}$ serves client $d \in \mathcal{D}$.
e_{wd}:	Lateness associated with truck $w \in \mathcal{W}$ and destination $d \in \mathcal{D}$.
x_{ia}:	Non-negative variables indicating the position of box $i \in \mathcal{B}$ in axis $a \in \mathcal{A}$.
$r^a_{ija}(r^b_{ija})$:	Binary variable indicating if box $i \in \mathcal{B}$ is after (before) $j \in \mathcal{B}$ in axis $a \in \mathcal{A}$, $i < j$.

$$
\begin{aligned}
&\sum_{f \in \mathcal{F}} \Delta_{kf} = 1, && k \in \mathcal{K}, && (2)\\
&\Delta_{kf} = \sum_{w \in \mathcal{W}_f} p_{iw}, && f \in \mathcal{F}, k \in \mathcal{K}, i \in \mathcal{B}_k, && (3)\\
&v_{wd} \geq p_{iw}, && d \in \mathcal{D}, i \in \mathcal{B}_d, w \in \mathcal{W}, && (4)\\
&\sum_{d \in \mathcal{D}} v_{wd} \leq 1, && w \in \mathcal{W}, && (5)\\
&m_w \geq (LT_w + \overline{u}_{if}) p_{iw}, && f \in \mathcal{F}, w \in \mathcal{W}_f, i \in \mathcal{B}, && (6)\\
&m_w + t_{fd} - \tau_d - (1 - v_{wd}) T_{dfw} \leq e_{wd}, && d \in \mathcal{D}, f \in \mathcal{F}, w \in \mathcal{W}_f, && (7)\\
&x_{ia} + \ell_{ia} \leq x_{ja} + (1 - r^{bf}_{ija}) L_a, && i, j \in \mathcal{B}, i < j, a \in \mathcal{A}, && (8)\\
&x_{ja} + \ell_{ja} \leq x_{ia} + (1 - r^{af}_{ija}) L_a, && i, j \in \mathcal{B}, i < j, a \in \mathcal{A}, && (9)\\
&\sum_{a \in \mathcal{A}} (r^{b}_{ija} + r^{a}_{ija}) \geq p_{iw} + p_{jw} - 1, && i, j \in \mathcal{B}, i < j, && (10)\\
&x_{ia} \leq L_a - \ell_{ia} && i \in \mathcal{B}, a \in \mathcal{A} && (11)\\
& && && (12)\\
& && && (13)\\
& && . && (14)\\
& && && (15)\\
& && && (16)
\end{aligned}
$$

The objective function (1) minimizes the cost of first and second layer distributions and the cost of delays. The first layer distribution cost is defined as the total cost of moving goods from origins $o \in \mathcal{O}$ to cross-docks $f \in \mathcal{F}$ and the second layer distribution costs is defined as the total costs of moving goods from cross-docks $f \in \mathcal{F}$ to respective destinations $d \in \mathcal{D}$. The third term is the cost of delays. Constraints (2) ensure that every shipment goes through exactly one of the cross-docks. In each cross-dock $f \in \mathcal{F}$, the inbound boxes are unloaded and loaded into one of the available trucks $\mathcal{W}_f$, according to constraints (3). The outbound trucks deliver goods to no more than one consumer (constraints (4) and (5)). The time each truck is ready for departure is defined by constraints (6), whilst delivery delays for each destination are computed by constraints (7) in which T_{dfw}, $d \in \mathcal{D}$, $f \in \mathcal{F}$, $w \in \mathcal{W}_f$ is a sufficiently big number. Each truck has a limited capacity, defined by its three dimensional geometry. Constraints (8) and (9) ensure there is no overlapping between any pair of boxes. Note that constraints (8) and (9) are only active if two boxes are in the same truck due to constraints (10). Constraints (11) define an upper bound for the position of the boxes. (12)(16)

Note that, since there are no constraints on the processing capabilities of the cross-docks $f \in \mathcal{F}$, we can compute the time from which each inbound box would become available for loading in facility f as $\overline{u}_{if} = t_{of} + UT_k + ST_k$, $o \in \mathcal{O}$, $k \in \mathcal{K}$, $i \in \mathcal{B}_k$, $f \in \mathcal{F}$.

One characteristic of the NSCD model (1)–(11) is that the number of variables and constraints grows with $O(|\mathcal{B}|^2)$. Also, the associated loading problem is know to be difficult to solve. The decomposition strategy presented in Section 4 scales down this effect by isolating each loading problem into a .

4 Decomposition method

The idea of a layered distribution can be explored to reformulate the NSCD model. In the described problem, we have two layers. For the first layer distribution, one needs to assign shipments to cross-docks. The second layer distribution consists of the dispatching of trucks to destinations. Let q_{fd} be a parameter indicating the optimal cost of distributing goods (plus delay penalties) in a subset of shipments $\mathcal{S} \subseteq \mathcal{K}$ from facility $f \in \mathcal{F}$ to destination $d \in \mathcal{D}$. Then, with non-negative variables z_{fd}, $f \in \mathcal{F}$, $d \in \mathcal{D}$, representing the minimum cost of performing the second layer distribution from facility f to destination d (plus delay penalties), we can reformulate model NSCD as (17), (2) and (18), which can be used in a Benders decomposition method as a master problem.

$$\text{(NSCD-B-MP) Min} \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} \sum_{f \in \mathcal{F}} c_{of} \Delta_{kf} + \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} z_{fd} \tag{17}$$

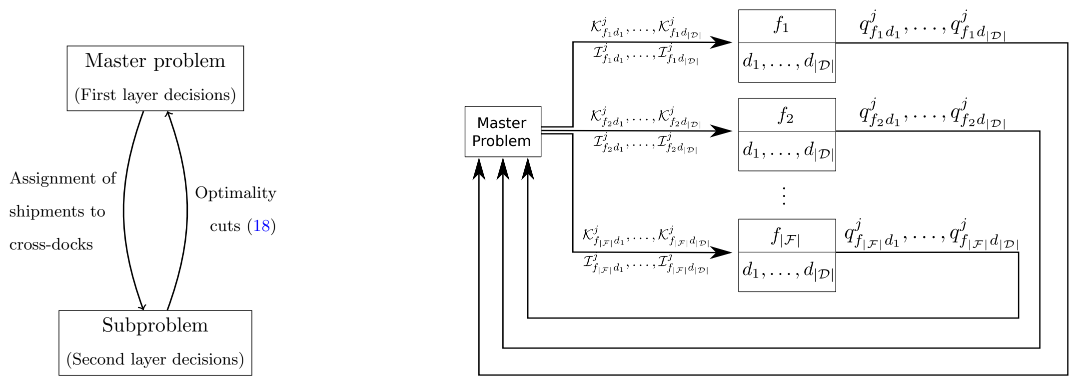
$$\sum_{f \in \mathcal{F}} \Delta_{kf} = 1, \qquad k \in \mathcal{K}, \tag{2}$$

$$z_{fd} \geq q_{fd} - q_{fd} \sum_{k \in \mathcal{S}} (1 - \Delta_{kf}), \quad f \in \mathcal{F}, d \in \mathcal{D}, \mathcal{S} \subseteq \mathcal{K} \tag{18}$$

$$\tag{19}$$

$$\tag{20}$$

Objective function (17) minimizes the total cost of first and second layer distribution and violating the deadlines. We re-use constraints (2), ensuring every shipment is sent to one cross-dock. Constraints (18) consider all the combinations of shipments k sent to a facility f that have at least one box for destination d and define the minimum second layer cost



(a) master .

(b)

Figure 3:

(with deadlines) for that particular combination. (31) (32)Instead of explicitly enumerating constraints (18), we can use a logic-based Benders decomposition algorithm to generating them iteratively.

The decomposition algorithm is summarized in Figure 3. At iteration $j = 1, \ldots, J$, we solve the relaxed master problem (NSCD-B-MP with some of constraints (18)) which gives a distribution plan for the first layer of the system, namely, the set of shipments that are sent to a facility f ($\Delta_{kf} = 1$) with at least one box for destination d, $\mathcal{K}^j_{fd}$ (which also define sets $\mathcal{I}^j_{fd}$, the set of boxes sent to facility f for destination d). Then, for each facility f, we have a that we can further decompose by destination, since goods for different destination cannot share a vehicle. Therefore, we solve the NSCD-B-Sub$_{fd}$ for each intermediate facility f and destination d.

$$\text{(NSCD-B-Sub}_{fd}\text{) Min } q^j_{fd} = \sum_{w \in \mathcal{W}_f} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} e_{wd}. \tag{21}$$

$$\sum_{w \in \mathcal{W}_f} p_{iw} = 1, \qquad i \in \mathcal{I}_{fd}. \tag{22}$$

$$m_w \geq (LT_w + \overline{u}_{if}) p_{iw}, \qquad w \in \mathcal{W}_f, i \in \mathcal{I}_{fd}, \tag{23}$$

$$m_w + t_{fd} - \tau_d - (1 - v_{wd})T \leq e_{wd}, \qquad w \in \mathcal{W}_f, \tag{24}$$

$$v_{wd} \geq p_{iw}, \qquad d \in \mathcal{D}, i \in \mathcal{B}_d, w \in \mathcal{W}, \tag{25}$$

$$\sum_{d \in \mathcal{D}} v_{wd} \leq 1, \qquad w \in \mathcal{W}_f, \tag{26}$$

$$x_{ia} + \ell_{ia} \leq x_{ja} + (1 - r_{ija}^b) L_a, \qquad i, j \in \mathcal{I}_{fd}, i < j, a \in \mathcal{A}, \tag{27}$$

$$x_{ja} + \ell_{ja} \leq x_{ia} + (1 - r_{ija}^a) L_a, \qquad i, j \in \mathcal{I}_{fd}, i < j, a \in \mathcal{A}, \tag{28}$$

$$\sum_{a \in \{1,2,3\}} (r_{ija}^b + r_{ija}^a) \geq p_{iw} + p_{jw} - 1, \qquad i, j \in \mathcal{I}_{fd}, i < j. \tag{29}$$

$$x_{ia} \leq L_a - \ell_{ia} \qquad i \in \mathcal{I}_{fd}, a \in \mathcal{A} \tag{30}$$

$$\tag{31}$$

$$. \tag{32}$$

$$\tag{33}$$

$$\tag{34}$$

Objective function (21) accounts for the distribution cost from cross-docks to destination plus delays. Constraints (22) ensure each box is loaded into a truck. The ready time of the truck is ensured by constraints (23). Constraints (24) compute the lateness of each truck. Each truck visits at most one destination, constraints (25) and (26)). The geometry of the loading is account for by constraints (27)–(30). (31)(34)

We remark that: (i) to ensure optimality of the solution of the decomposition method, we require an optimal solution of the at least in the last step. However, the method can still provide feasible solutions when using a lower bound for the ; (ii) if we cannot find a feasible solution in reasonable time and the cut generated by the lower bound is not enough to cut off the current solution, the method stops with a sub-optimal solution.

4.1

The main drawback of constraints (18) is that changing the value of just one of variables Δ_{kf} is enough to all the information gained in the previous iteration of the algorithm. This effect can be reduced. Note that q_{fd} consists of the aggregated costs of each shipment used to load the truck going to a particular destination. Therefore, whenever the value of a variable Δ_{kf} is changed from one to zero, we want to discount from q_{fd} only the part of the cost associated with shipment k. For this, we can solve a of distributing good from shipment k to all respective destinations, producing the cost (or an upper bound) M_{kfd} of distribution goods from k through f to d. Allowing us to replace (18) by (35) in NSCD-B-MP.

$$z_{fd} \geq q_{fd} - \sum_{k \in \mathcal{S}} M_{kfd}(1 - \Delta_{kf}), \quad f \in \mathcal{F}, d \in \mathcal{D}, \mathcal{S} \subseteq \mathcal{K}. \tag{35}$$

5 Computational experiments

All the computational experiments were run using an Intel Xeon CPU E5-2620@2.0GHz and 64GB of RAM. Regarding software, we used Gurobi 9.0.1. The implementation of the Benders algorithm was done using the callback feature. The source code for the NSCD model, the Benders decomposition and the generation of instances is available as supplementary material.

In Subsection 5.1, we describe the generation of instances. These instances were used for the evaluation of the NSCD model, see Subsection 5.2, and for the comparison of the NSCD model against the Benders decomposition strategy (Subsection 5.3). Detailed results for individual instances is also available as supplementary material.

5.1 Generating the instances

Since there are no instances for the *network scheduling Problem with cross-docking and loading constraints* problem, we generated them based on instances for the vehicle routing problem (Augerat et al., 1995) and on a test set for the container loading problem (Ivancic et al., 1989). From the vehicle routing instances, we sampled the position of the nodes in the network and the geometry of the trucks and boxes were sampled from the container loading test set.

We generate eight sets of instances with different combinations of the number of nodes ($|\mathcal{O}| + |\mathcal{F}| + |\mathcal{D}|$) and the total number of boxes ($|\mathcal{B}|$), see Table 3. For each combination of parameters, we choose randomly the specified number of points and randomly select one of the instances of the container loading problem. Then, the determined number of boxes is sampled from the types of boxes in the container loading instance. The status of origin (supplier), cross-dock or destination (consumer) is assigned randomly ensuring the number of cross-docks is between two and 20% of the nodes. The boxes are assigned randomly to suppliers and destinations so that each supplier and each destination has at least one box. We assume the velocity of the vehicle is unitary so distance and time are numerically the same. The unloading, sorting and loading times are each computed as one-third of the minimum distance between any pair of nodes. To generate the deadlines, we consider the interval $[\overline{\tau}_d^L, \overline{\tau}_d^H]$ in which $\overline{\tau}_d^L = \min_{i\in\mathcal{B}, f\in\mathcal{F}}\{\overline{u}_{if} + LT + c_{fd}\}$ and $\overline{\tau}_d^H = \max_{i\in\mathcal{B}, f\in\mathcal{F}}\{\overline{u}_{if} + LT + c_{fd}\}$. Then, we sampled a value from $[1.1\overline{\tau}_d^L, 0.9\overline{\tau}_d^H]$ to be the center of the time window for the consumer and we sampled the width of a time window to be from 30% to 60% of $[\overline{\tau}_d^L, \overline{\tau}_d^H]$, then the upper bound of the time window is set to be the deadline τ_d, $d \in \mathcal{D}$. All the samples over intervals are from a uniform distribution. This procedure is repeated for each of the 27 instances of the A-set from Augerat et al. (1995) for each combination of number of nodes and boxes.

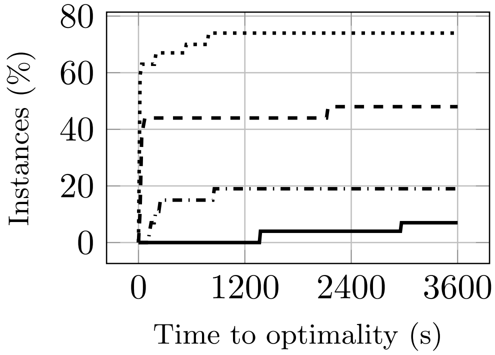
Table 3: Set of instances for network scheduling problem with cross-docking.

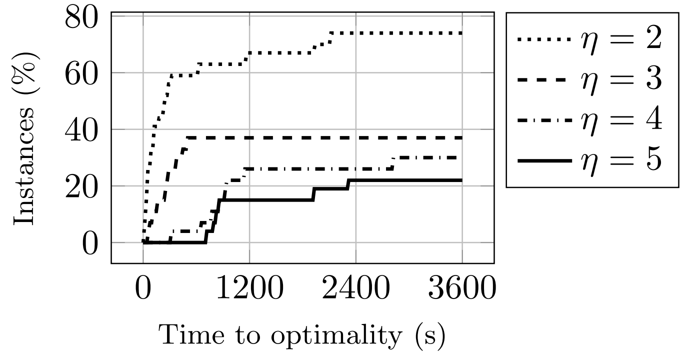
	Set 1	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Set 6**	**Set 7**	**Set 8**
Nodes	10	10	10	10	20	20	20	20
Boxes	20	40	60	100	40	60	80	100

5.2 Evaluating the NSCD model

Regarding the solver performance when using model NSCD, (1)–(10), metric yields a rough estimate on the relative difficulty in solving the instance, this phenomenon can be seen when observing the time to solve each instance or the gap of the instances after 3600 seconds (time limit, TL). profile of computational time and gap for 10 and 20 nodes in the network, respectively, for different η. Detailed results for individual instances are provided as supplementary material.

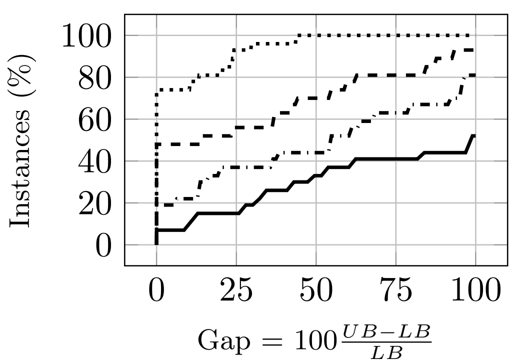
The profiles for the computational time for each instance (with a time limit of an hour) show that the higher the value of η the longer it takes to solve the instance on average. Note that the total size of the problem (number of variables and constraints) is not necessarily a

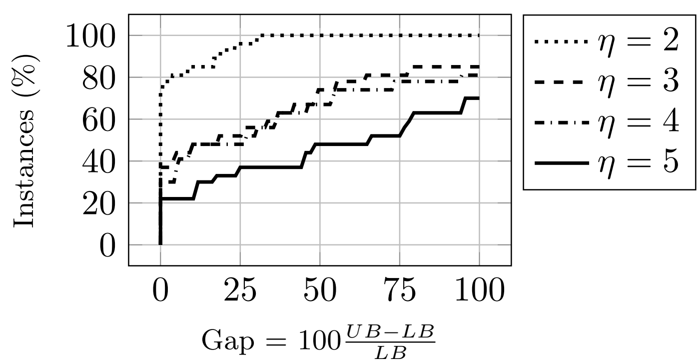




(a) time horizontal axisfor instances with 10 nodes.

(b) time horizontal axisfor instances with 20 nodes.





(c)

(d)

Figure 4:

goods predictor of performance. Comparing the profiles in figures 4a and 4b, for example, we can see that instances with 10 nodes and 40 boxes ($\eta = 4$, Figure 4a) took longer to solve than instances with 20 nodes and 40 boxes ($\eta = 2$, Figure 4b).

The same phenomenon can be observed comparing the profiles in figures 4c and 4d. Higher values of η lead to higher gaps. Again, we can see that bigger problems are not necessarily harder to solve. For example, after an hour, around 50% of instances with 10 nodes and and 100 boxes ($\eta = 10$, Figure 4c) achieved a gap lower than 100% whilst around 70% of instances with 20 nodes and 100 boxes ($\eta = 5$, Figure 4d) reached the same level of gap.

5.3 Comparing NSCD model and the decomposition algorithm

To evaluate the decomposition algorithm proposed in Section 4, we compared it to the NSCD model (Section 3) regarding the quality of lower (dual) and upper (primal) bounds provided be each of them and regarding the computational time. The time limit was set to 3600 seconds for the master problem with 1% of that value for each . For the decomposition algorithm, we consider the relaxed master problem (NSCD-B-MP), changing constraints (18) by (35), with constraints (35) such that $|\mathcal{S}| = 1$ added *a priori*.

Specially for instances with bigger values of η, the decomposition algorithm is more effective in providing better bounds. For the smaller instances (set 1, Figure 5), the decomposition proved optimal solutions for all of the instances, which could not be done in seven instances with Gurobi's branch-and-cut applied to the NSCD model. As the instances get larger, the difference in performance favors the decomposition (see, for example, 3, 4, 13, 22, 23, 26 and 27 in Figure 6). The other sets of instances had a similar behavior (see figures 7, 8, 9, 10 11 and 12 in the Appendix).

Median results for time to solve (in seconds) and gap also confirm the decomposition algorithm is more effective for the different sets of instances (see Table 4). We can see that for all but the sets with smallest η (sets 1 and 5) at least half of the instances reached the time limit for the Gurobi's branch-and-cut whilst the decomposition algorithm had the highest median result in set 8 with a value less than 25% of the time limit of 3600 seconds. This performance difference is also observed in the median gap in which, again, Gurobi's procedure could only prove optimality of half of the instance for the smallest sets. This is

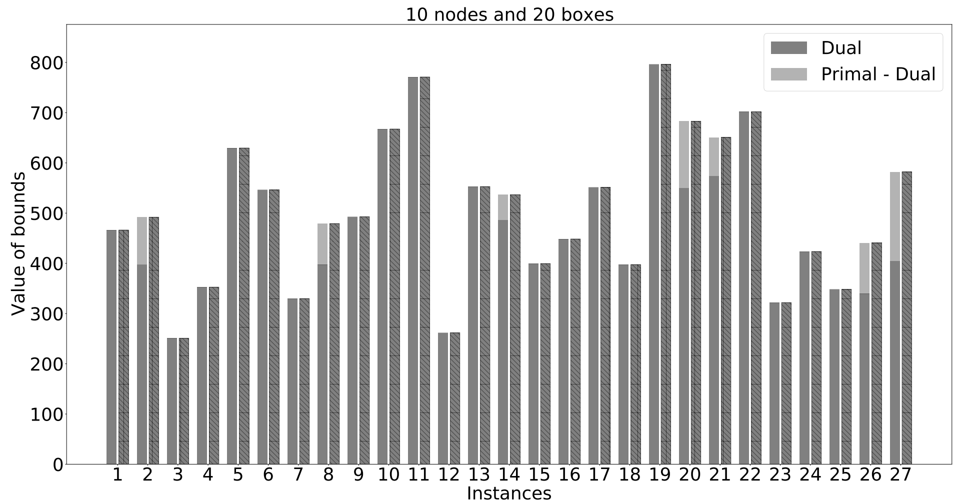


Figure 5: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 20 boxes (set 1).

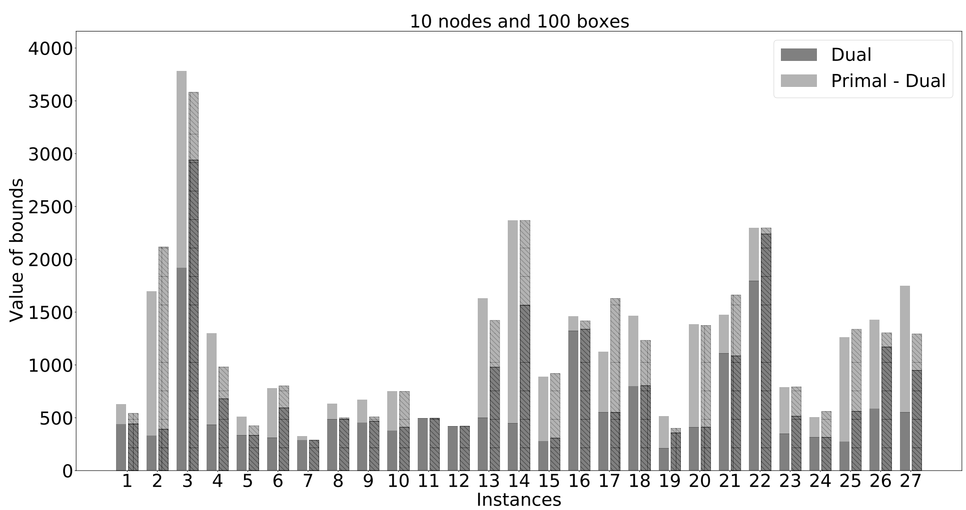


Figure 6: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 100 boxes (set 4).

reflected in the number of optimal solutions found in which the decomposition outperforms the NSCD in every set.

Table 4: Summary of the median results for time to solve (seconds) and gap $(100\frac{UB-LB}{LB})$ for NSCD and the decomposition algorithm. It also shows the number of optimal solutions found for each set of instances. TL indicates time limit reached (one hour).

	Time		**Gap**		**# Optimals**	
	NSCD	**Decomposition**	**NSCD**	**Decomposition**	**NSCD**	**Decomposition**
Set 1	4	3	0.00	0.00	20	27
Set 2	TL	15	14.02	0.00	13	19
Set 3	TL	342	54.76	13.45	5	10
Set 4	TL	701	97.55	36.37	2	3
Set 5	236	6	0.00	0.00	20	23
Set 6	TL	483	18.20	0.00	10	18
Set 7	TL	66	26.88	0.00	8	15
Set 8	TL	875	65.81	1.55	6	13

The difference in gap is better explained by the improvement in the dual bound rather than the primal bound. The NSCD provided a better primal bound than the decomposition in 33 instances and the decomposition provided a better primal bound in 49 instances. For the dual bound, on the other hand, the decomposition provided a better value in 87 instances (against 23 of the NSCD). This is consistent in all bigger sets of instance (see Table 5) and for time to solve and gap as well. The highest difference is observed in time to solve in which the decomposition was better 178 cases (against 11 of the NSCD).

results for each instance

9 and 7 in the Appendix.

Thereby, the experiments revealed the efficacy of the decomposition algorithm in providing good primal and dual bounds with considerable savings in computational time. This is particularly interesting for increasing the applicability of the *network scheduling problem with cross-docking and loading constraints* for larger instances. One possible explanation for this efficacy is the reduction in size of the resulting loading problems in the decomposition.

Table 5:

# Wins	Time		Gap		Primal		Dual	
	NSCD	Decomp.	NSCD	Decomp.	NSCD	Decomp.	NSCD	Decomp.
Set 1	6	21	0	7	0	0	0	7
Set 2	0	27	0	13	2	0	0	13
Set 3	0	27	5	15	7	7	0	13
Set 4	0	25	4	21	7	15	1	18
Set 5	3	22	2	5	2	1	2	5
Set 6	2	20	3	12	4	7	2	12
Set 7	0	18	9	10	6	7	9	10
Set 8	0	18	9	12	5	12	9	9
Total	**11**	**178**	**32**	**95**	**33**	**49**	**23**	**87**

6 Conclusions and future work

We the *network scheduling problem with cross-docking and loading constraints* in which the goal is to route goods in a two-layered open network so that goods from suppliers reach consumers through consolidation centers (cross-docks) minimizing transportation and delay costs. The demands consider the geometry of the goods, not just their volume. Even though the vehicle routing literature contemplates cases with three-dimensional loading constraints this is the first time a network distribution problem with cross-docking considers this feature, to the best of our knowledge. The resulting model might be challenging to solve for instances of moderate size, for such cases, we propose an effective Benders based decomposition strategy enhancing the applicability of the problem to larger instances.

The network scheduling problem with cross-docking and loading constraints has some interesting that can be explored in the future. Among then, it is possible to consider the case in which goods are irregular objects which could demand more procedures for solving the correspondent subproblems. Another interesting case is allowing for multiple visits on the second layer distribution (with or without the vehicle returning to the cross-dock of origin). . Furthermore, .

Table 6:

	10 nodes and 20 boxes					10 nodes and 40 boxes				
	NSCD		**Decomposition**			**NSCD**		**Decomposition**		
Inst.	**Time**	**Gap**	**Time**	**Gap**	**Constr.**	**Time**	**Gap**	**Time**	**Gap**	**Constr.**
1	1	0.00	< 1	0.00	34	TL	160.92	1508	58.09	74
2	TL	23.66	30	0.00	122	69	0.00	8	0.00	28
3	1	0.00	< 1	0.00	30	23	0.00	10	0.00	72
4	< 1	0.00	3	0.00	57	9	0.00	1	0.00	42
5	2	0.00	3	0.00	83	TL	83.90	563	26.53	81
6	2	0.00	< 1	0.00	38	23	0.00	< 1	0.00	36
7	4	0.00	1	0.00	42	20	0.00	1	0.00	44
8	TL	20.64	5	0.00	123	38	0.00	1	0.00	50
9	7	0.00	3	0.00	89	27	0.00	5	0.00	34
10	184	0.00	4	0.00	62	31	0.00	13	0.00	84
11	522	0.00	10	0.00	163	TL	134.50	521	125.19	30
12	< 1	0.00	3	0.00	140	12	0.00	1	0.00	48
13	776	0.00	< 1	0.00	62	TL	62.08	3	0.00	83
14	TL	10.31	30	0.00	80	TL	14.02	170	0.00	96
15	1	0.00	< 1	0.00	35	TL	43.72	16	0.00	129
16	< 1	0.00	2	0.00	72	5	0.00	< 1	0.00	48
17	26	0.00	< 1	0.00	72	TL	54.28	15	0.00	120
18	2	0.00	2	0.00	28	TL	36.85	171	0.00	108
19	4	0.00	1	0.00	76	TL	36.58	234	6.48	110
20	TL	24.22	12	0.00	67	35	0.00	34	0.00	158
21	TL	13.33	7	0.00	201	TL	87.14	724	28.42	75
22	1	0.00	7	0.00	251	2128	0.00	2	0.00	48
23	1	0.00	< 1	0.00	39	TL	92.72	93	0.00	116
24	12	0.00	4	0.00	38	27	0.00	2	0.00	60
25	< 1	0.00	< 1	0.00	36	TL	24.11	426	3.17	82
26	TL	29.71	44	0.00	60	TL	59.14	623	8.70	118
27	TL	43.73	5	0.00	142	TL	42.36	435.56	42.36	22
Median	**4.14**	**0.00**	**2.82**	**0.00**		**TL**	**14.02**	**15.12**	**0.00**	

Table 7: () gap ($100\frac{UB-LB}{LB}$) and is .

	20 nodes and 80 boxes					**20 nodes and 100 boxes**				
	NSCD		**Decomposition**			**NSCD**		**Decomposition**		
Inst.	**Time**	**Gap**	**Time**	**Gap**	**Constr.**	**Time**	**Gap**	**Time**	**Gap**	**Constr.**
1	TL	73.40	TL	96.22	3117	TL	143.92	14	0.00	112
2	2823	0.00	3	0.00	132	TL	47.23	69	0.00	706
3	848	0.00	3	0.00	2223	TL	294.17	TL	338.62	552
4	TL	35.53	TL	60.76	210	783	0.00	10	0.00	310
5	772	0.00	66	0.00	6754	TL	99.18	TL	147.70	752
6	927	0.00	4	0.00	428	TL	94.83	16	0.00	251
7	TL	26.88	4	0.00	546	TL	11.12	48	0.00	300
8	TL	4.74	TL	138.21	8950	TL	65.81	125	0.00	1737
9	645	0.00	160	0.00	578	TL	78.51	1627	157.02	78
10	TL	36.96	TL	108.02	299	TL	75.12	47	0.00	525
11	319	0.00	35	0.00	4322	TL	110.71	361	0.00	1317
12	TL	10.11	TL	102.67	7029	TL	17.00	TL	143.97	2192
13	TL	45.91	18	0.00	5354	TL	23.72	TL	41.94	520
14	TL	5.83	4	0.00	65	TL	193.10	2147	1.55	2977
15	TL	137.04	912	2.31	420	1923	0.00	129	0.00	1342
16	TL	8.86	8	0.00	389	TL	125.67	2605	196.21	124
17	TL	54.09	53	0.00	352	702	0.00	495	0.00	1571
18	TL	94.25	1771	115.95	62	TL	77.57	875	13.48	1343
19	TL	29.79	2404	18.50	1266	TL	45.37	2512	11.13	1198
20	TL	4.35	TL	41.71	460	TL	94.40	TL	69.72	696
21	TL	168.73	TL	221.41	2466	TL	254.73	TL	338.29	692
22	TL	148.03	34	0.00	3764	2310	0.00	3	0.00	159
23	TL	116.36	TL	136.77	446	846	0.00	3	0.00	137
24	1134	0.00	5	0.00	3232	TL	11.06	TL	10.45	652
25	934	0.00	4	0.00	2256	TL	208.53	TL	265.20	615
26	TL	123.77	27	0.00	569	825	0.00	2	0.00	128
27	TL	53.86	TL	48.13	891	TL	44.48	TL	51.90	1979
Median	**TL**	**26.88**	**65.80**	**0.00**		**0.00**	**65.81**	**874.55**	**1.55**	

Acknowledgments

We thank the State of Sao Paulo Research Foundation (FAPESP), process numbers, #2017/01097-9, #2015/15024-8, CEPID #2013/07375-0 and the National Council for Scientific and Technological Development (CNPq), grant #308761/2018-9 from Brazil.

References

D. Agustina, C. Lee, and R. Piplani. A Review: Mathematical Modles for Cross Docking Planning. *International Journal of Engineering Business Management*, 2:47–54, 2010.

P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. *Computational results with a branch and cut code for the capacitated vehicle routing problem*, volume 34. IMAG, 1995.

J. V. Belle, P. Valckenaers, and D. Cattrysse. Cross-docking: State of the art. *Omega*, 40: 827–846, 2012.

A. Bortfeldt and G. Wäscher. Constraints in container loading – A state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20, 2013.

A. Bortfeldt and J. Yi. The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints. *European Journal of Operational Research*, 282(2):545–558, 2020.

P. Buijs, I. F. A. Vis, and H. J. Carlo. Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239: 593–608, 2014.

S. Ceschia, A. Schaerf, and T. Stützle. Local search techniques for a routing-packing problem. *Computers and Industrial Engineering*, 66:1138–1149, 2013.

P. Chen, Y. Guo, A. Lim, and B. Rodrigues. Multiple crossdocks with inventory and time windows. *Computers and Operations Research*, 33:43–63, 2006.

R. L. Cook, B. Gibson, and D. MacCurdy. A lean approach to cross docking. *Supply Chain Management review*, pages 54–59, 2005.

A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32:1429–1450, 2005.

G. Forger. UPS starts world's premiere cross-docking operation. *Modern Material Handling*, 36:36–38, 1995.

S. Gelareh, F. Glover, O. Guemri, S. Hanafi, P. Nduwayo, and R. Todosijević. A comparative study of formulations for a cross-dock door assignment problem. *Omega*, 91:102015, 2020.

M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350, 2006.

A. M. Geoffrion and G. W. Graves. Multicommodity Distribution System Design by Benders Decomposition. *Management Science*, 20:822–844, 1974.

G. Guastaroba, M. G. Speranza, and D. Vigo. Intermediate Facilities in Freight Transportation Planning : A Survey. *Transportation Science*, 50:763–789, 2016.

S. Hernández, S. Peeta, and G. Kalafatas. A less-than-truckload carrier collaboration planning problem under dynamic capacities. *Transportation Research Part E: Logistics and Transportation Review*, 47:933–946, 2011.

P. Hokama, F. K. Miyazawa, and E. C. Xavier. A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, 47:1–13, 2016.

J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96:33–60, 2003.

N. J. Ivancic, K. Mathur, and B. B. Mohanty. An integer-programming based heuristic approach to the three-dimensional packing problem. *Journal of Manufacturing & Operations Management*, 2:268–298, 1989.

L. Junqueira, J. F. Oliveira, M. A. Carravilla, and R. Morabito. An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research*, 20(5):645–666, 2013.

E. Kinnear. Is there any magic in cross-docking? *Supply Chain Management: An International Journal*, 2:49–52, 2006.

H. Koch, M. Schlögell, and A. Bortfeldt. A hybrid algorithm for the vehicle routing problem with three-dimensional loading constraints and mixed backhauls. *Journal of Scheduling*, 23(1):71–93, 2020.

D. Männel and A. Bortfeldt. A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. *European Journal of Operational Research*, 254(3):840–858, 2016.

F. Massen, Y. Deville, and P. Van Hentenryck. Pheromone-based heuristic column generation for vehicle routing problems with black box feasibility. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 260–274. Springer, 2012.

A. Moura. A model-based heuristic to the vehicle routing and loading problem. *International Transactions in Operational Research*, 26(3):888–907, 2019.

M. Napolitano. Cross dock fuels growth at Dots. *Logistics Management*, 50:54–9, 2011.

H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330, 2015.

S. Reil, A. Bortfeldt, and L. Mönch. Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints. *European Journal of Operational Research*, 266(3):877–894, 2018.

G. Stalk, P. Evans, and L. E. Shulman. Competing on capabilities: the new rules of corporate strategy. *Harvard Business Review*, 70:57–69, 1992.

K. Stephan and N. Boysen. Cross-docking. *Journal of Management Control*, 22:129–137, 2011.

C. A. Vega-Mejía, J. R. Montoya-Torres, and S. M. Islam. Consideration of triple bottom line objectives for sustainability in the optimization of vehicle routing and loading operations: a systematic literature review. *Annals of Operations Research*, 273(1-2):311–375, 2019.

C. Witt. Crossdocking: concepts demand choice. *Material Handling Engineering*, 53:44–49, 1998.

V. F. Yu, P. Jewpanya, and A. A. N. P. Redi. Open vehicle routing problem with cross-docking. *Computers and Industrial Engineering*, 94:6–17, 2016.

X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowsland. A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, 23(1-2):287–320, 2016.

W. Zhu, H. Qin, A. Lim, and L. Wang. A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers and Operations Research*, 39(9):2178–2195, 2012.

Appendix

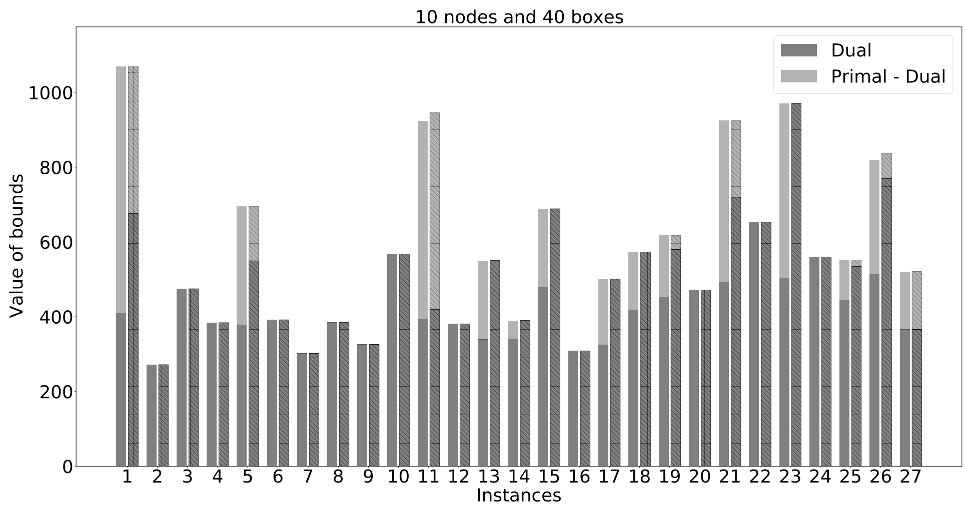


Figure 7: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 40 boxes (set 2). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

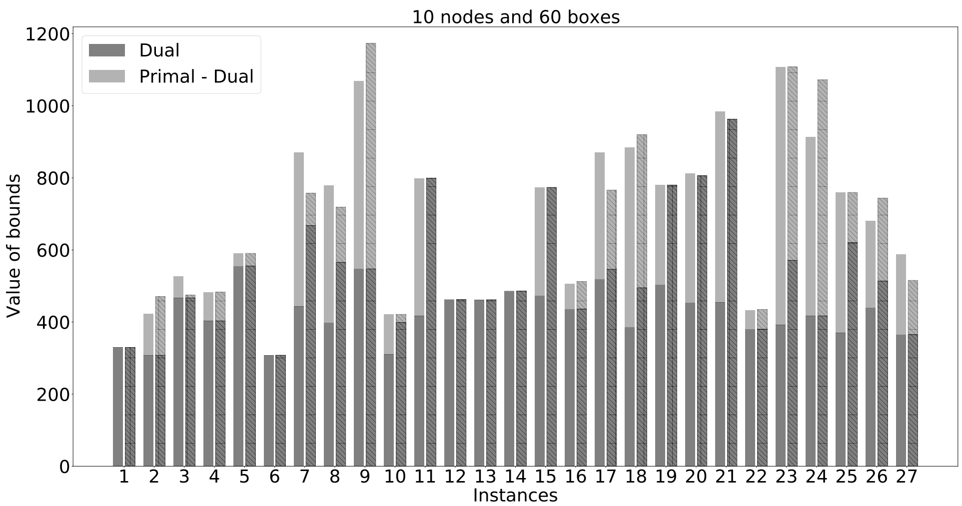


Figure 8: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 60 boxes (set 3). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

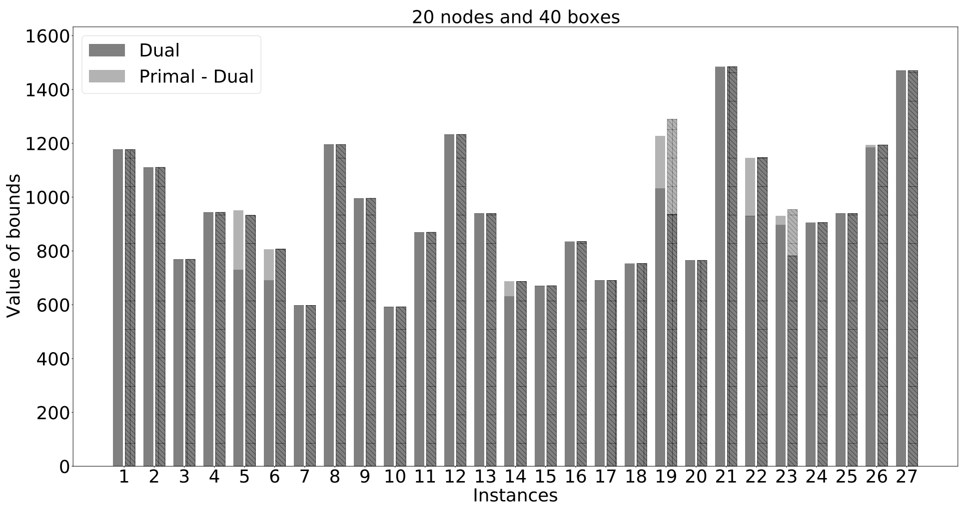


Figure 9: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 20 nodes and 40 boxes (set 5). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

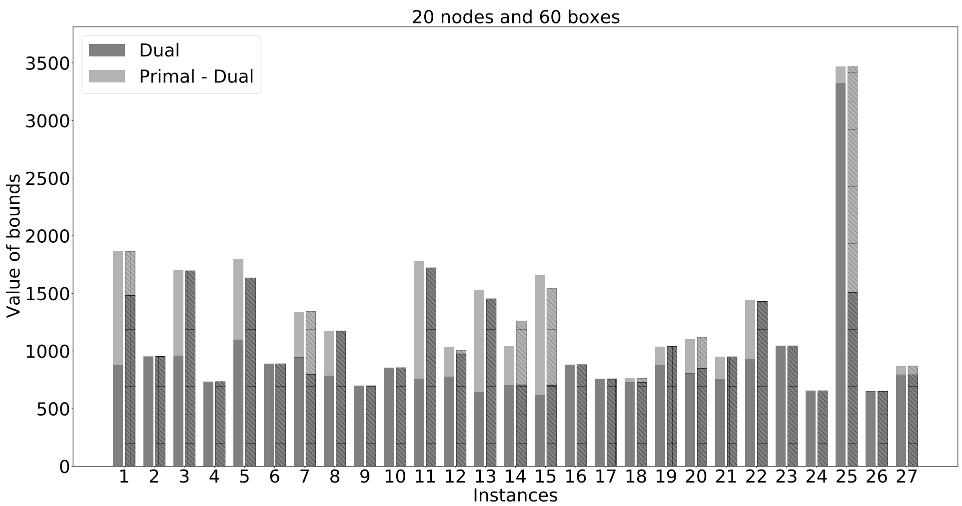


Figure 10: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 60 boxes (set 6). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

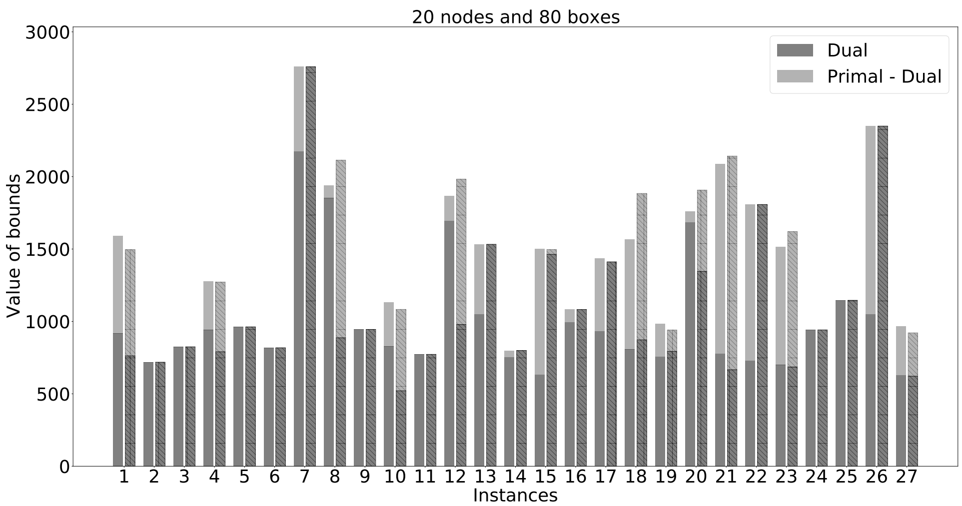


Figure 11: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 80 boxes (set 7). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

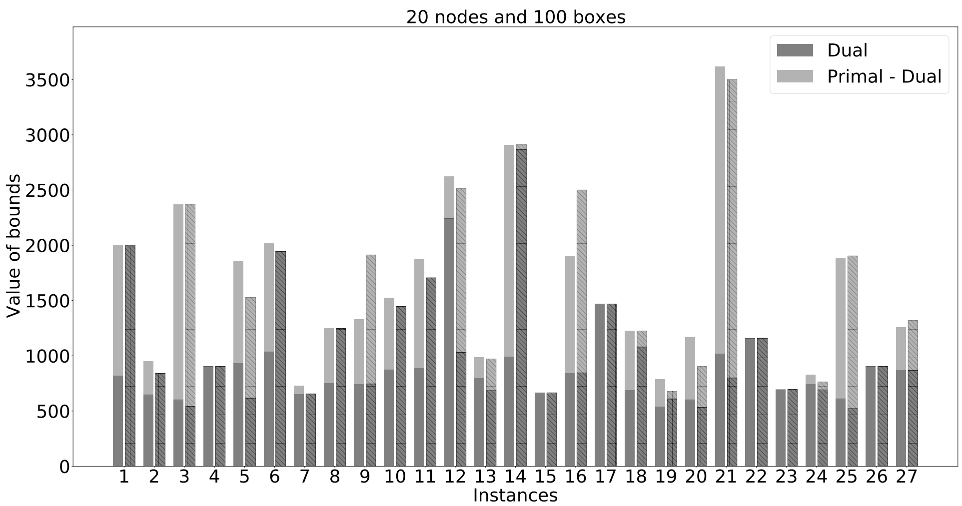


Figure 12: Representation of primal (upper) and dual (lower) bounds for the NSCD model and the Benders decomposition for the case with 10 nodes and 100 boxes (set 8). For each instance (1-27, x-axis), there are two bars, the one on the left shows dual bound (dark) and the absolute gap (light) when using the NSCD model, the one on the right shows the same metric when using the Benders decomposition algorithm.

Table 8: Computational time (in seconds) and gap ($100\frac{UB-LB}{LB}$) for each instance in cases with 10 nodes and 60 boxes (set 4) and 10 nodes and 100 boxes (set 5). For the decomposition algorithms, the number of generated constraints is also counted.

	10 nodes and 60 boxes					**10 nodes and 100 boxes**				
	NSCD		**Decomposition**			**NSCD**		**Decomposition**		
Inst.	**Time**	**Gap**	**Time**	**Gap**	**Constr.**	**Time**	**Gap**	**Time**	**Gap**	**Constr.**
1	841	0.00	28	0.00	39	TL	41.89	320	23.17	48
2	TL	37.56	130	53.15	58	TL	412.93	1069	440.49	46
3	TL	13.02	342	1.71	90	TL	97.06	4614	21.79	224
4	TL	19.86	343	19.86	34	TL	200.78	672	44.04	42
5	TL	6.37	901	6.37	202	TL	52.50	321	27.46	52
6	189	0.00	2	0.00	30	TL	151.40	327	35.27	42
7	TL	96.32	571	13.45	108	TL	12.65	19	0.00	36
8	TL	95.60	241	27.03	50	TL	31.02	2620	2.95	216
9	TL	95.16	815	114.35	40	TL	47.70	2072	8.55	250
10	TL	35.74	432	5.72	48	TL	97.55	655	82.36	42
11	TL	91.59	287	0.00	136	2960	0.00	10	0.00	42
12	126	0.00	12	0.00	72	1376	0.00	6	0.00	36
13	126	0.00	4	0.00	65	TL	227.11	1581	45.29	116
14	244	0.00	3	0.00	54	TL	430.31	404	51.44	42
15	TL	63.40	7	0.00	48	TL	216.44	781	198.41	39
16	TL	16.21	2403	17.66	192	TL	10.56	313	5.92	65
17	TL	67.63	1905	40.19	100	TL	104.25	1436	196.44	56
18	TL	129.09	2228	85.92	98	TL	83.57	TL	53.36	170
19	TL	54.76	175	0.00	80	TL	142.47	701	12.56	60
20	TL	79.19	250	0.00	100	TL	237.22	2436	233.94	82
21	TL	116.39	92	0.00	60	TL	32.91	2263	53.08	90
22	TL	13.66	704	14.44	70	TL	27.91	360	2.58	148
23	TL	182.60	1460	94.03	75	TL	127.13	318	53.64	45
24	TL	119.22	813	157.05	42	TL	60.63	757	78.25	34
25	TL	105.21	234	22.45	60	TL	357.50	1822	138.68	88
26	TL	54.65	2254	44.77	100	TL	143.46	643	11.64	48
27	TL	61.11	1520	41.29	66	TL	216.95	724.92	36.37	60
Median	**TL**	**54.76**	**341.88**	**13.45**		**TL**	**97.55**	**700.54**	**36.37**	

Table 9: Computational time (in seconds) and gap ($100\frac{UB-LB}{LB}$) for each instance in cases with 20 nodes and 40 boxes (set 5) and 20 nodes and 60 boxes (set 6). For the decomposition algorithms, the number of generated constraints is also counted.

	20 nodes and 40 boxes					**20 nodes and 60 boxes**				
	NSCD		**Decomposition**			**NSCD**		**Decomposition**		
Inst.	**Time**	**Gap**	**Time**	**Gap**	**Constr.**	**Time**	**Gap**	**Time**	**Gap**	**Constr.**
1	2108	0.00	3	0.00	529	TL	112.02	TL	25.76	3117
2	611	0.00	36	0.00	2600	361	0.00	1	0.00	132
3	42	0.00	2	0.00	432	TL	77.71	66	0.00	2223
4	1926	0.00	< 1	0.00	129	488	0.00	2	0.00	210
5	TL	30.14	1323	0.00	1564	TL	64.44	786	0.00	6754
6	TL	16.60	5	0.00	402	438	0.00	578	0.00	428
7	21	0.00	3	0.00	344	TL	41.25	TL	68.11	546
8	110	0.00	1	0.00	258	TL	49.15	589	0.00	8950
9	236	0.00	4	0.00	569	74	0.00	11	0.00	578
10	13	0.00	64	0.00	1123	263	0.00	4	0.00	299
11	1166	0.00	8	0.00	804	TL	133.98	483	0.00	4322
12	37	0.00	< 1	0.00	86	TL	33.47	3592	3.24	7029
13	41	0.00	18	0.00	458	TL	137.54	876	0.00	5354
14	TL	8.55	73	0.00	678	TL	47.90	1336	78.79	65
15	72	0.00	7	0.00	671	TL	167.86	TL	119.08	420
16	114	0.00	2	0.00	207	163	0.00	4	0.00	389
17	311	0.00	3	0.00	357	239	0.00	4	0.00	352
18	27	0.00	115	0.00	2876	TL	4.74	1170	4.74	62
19	TL	18.76	TL	37.72	796	TL	18.20	22	0.00	1266
20	9	0.00	53	0.00	2289	TL	36.90	TL	32.08	460
21	79	0.00	1	0.00	231	TL	25.35	463	0.00	2466
22	TL	23.27	1	0.00	124	TL	54.94	81	0.00	3764
23	TL	3.68	TL	22.05	708	95	0.00	4	0.00	446
24	228	0.00	2	0.00	309	279	0.00	2365	0.00	3232
25	176	0.00	114	0.00	5824	TL	4.19	TL	129.67	2256
26	TL	0.71	85	0.00	3709	59	0.00	6	0.00	569
27	280	0.00	6	0.00	946	TL	9.65	372.15	9.65	891
Median	**236.20**	**0.00**	**5.52**	**0.00**		**0.00**	**18.20**	**569.00**	**0.00**	