Proofs of statements and additional details about the computational experiments

EC.1. Proof of statements

EC.1.1. Proof of Proposition 1

The matrix associated with constraints (12b), (12c) and (12d) has coefficients in $\{0,1,-1\}$ and each column of its transpose has exactly two nonzero elements of a different sign. Hence, it is totally unimodular and, together with the bound constraints (12e)-(12i), ensures that the extreme points of (12b)-(12i) are integral (see, for example Bertsimas and Weismantel 2005).□

EC.1.2. Proof of Theorem 1

Let $(\bar{x},\bar{y},\bar{w})$ be an optimal solution of formulation SND-HC($\mathcal{D}^{\dagger}$) (i.e., an optimal CTSNDP-HC solution) of cost $\bar{z}$, and let $\overline{\mathcal{A}}=\{((i,t),(j,t+\tau_{ij}))\in\mathcal{A}_{\hat{\mathcal{T}}}:\overline{y}_{ij}^{t,t+\tau_{ij}}>0\}$ be the set of arcs traversed by the commodities. Below we show that to solution $(\bar{x},\bar{y},\bar{w})$ corresponds a feasible, but not necessarily optimal, solution (x,y,w) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of cost $z=\bar{z}$.

By means of the mapping described by expressions (13), we can associate with vectors $\bar{x}$ and $\bar{y}$ and corresponding paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$, solution vectors x and y. As shown by Boland et al. (2017), to solution vector $\bar{x}$ corresponds a set $\{P^k\}_{k\in\mathcal{K}}$ of feasible paths in network $\mathcal{D}_{\mathcal{T}}$, one path for each commodity, with the same total fixed and flow cost of paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$. More precisely, for each commodity $k\in\mathcal{K}$ and path $\overline{P}^k=(a_1^k,\ldots,a_{\eta^k}^k)$, $a_h^k\in\overline{\mathcal{A}}$, $h=1,\ldots,\eta^k$, with $a_h^k=((i_h^k,t_h^k),(i_{h+1}^k,t_h^k+\tau_{i_h^k i_{h+1}^k}))$ and $t_{h+1}^k\geq t_h^k+\tau_{i_h^k i_{h+1}^k}$ for $h=1,..,\eta^k-1$ induced by solution vector $\bar{x}$, corresponds a feasible path $P^k=(\mu(a_1^k),\ldots,\mu(a_{\eta^k}^k))$ in $\mathcal{D}_{\mathcal{T}}$ with appropriate holding arcs. For each $k\in\mathcal{K}$ we have that the total transit time $T(P^k)$ of path P^k can be computed as

$$T(P^k)=T(\overline{P}^k)=\sum_{h=1}^{\eta^k}\tau_{i_h^k i_{h+1}^k}=\sum_{((i,t),(j,\bar{t}))\in\overline{\mathcal{A}}}\tau_{ij}\overline{x}_{ij}^{kt\bar{t}}=\sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}}\tau_{ij}x_{ij}^{kt\bar{t}}. \tag{EC.1}$$

The holding times $\bar{w}$ of solution $(\bar{x},\bar{y},\bar{w})$ can be computed as:

$$\overline{w}_i^k=\begin{cases}t_1^k-e^k, & i=o^k,\\ l^k-(t_{\eta^k}^k+\tau_{i_{\eta^k}^k d^k}), & i=d^k,\\ t_h^k-(t_{h-1}^k+\tau_{i_{h-1}^k i_h^k}), & i=i_h^k, h=2,...,\eta^k,\\ 0, & \text{otherwise},\end{cases},\forall\ i\in\mathcal{N},\forall\ k\in\mathcal{K}.$$

It is easy to see that for each $k\in\mathcal{K}$ we have

$$\sum_{i\in\mathcal{N}}\overline{w}_i^k=l^k-e^k-T(\overline{P}^k). \tag{EC.2}$$

We now show that solution $w_i^k=\overline{w}_i^k$, $\forall i\in\mathcal{N}$, $k\in\mathcal{K}$, satisfies constraints (19)-(20), thus showing that to solution $(\bar{x},\bar{y},\bar{w})$ corresponds a feasible solution (x,y,w) of SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of the same

total fixed, flow and holding cost. For each path P^k, $k \in \mathcal{K}$, we have that equation (20) follows from expression (EC.1) and (EC.2). Constraints (19) for the values w_i^k and path P^k are as follows:

$$w_i^k \leq \begin{cases} \xi^k(\mu(a_1^k)) - e^k, & i = o^k, \\ l^k - \psi^k(\mu(a_{\eta^k}^k)), & i = d^k, \\ \xi^k(\mu(a_h^k)) - \psi^k(\mu(a_{h-1}^k)), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.3}$$

where for each a_h^k, $h = 1, ..., \eta^k$,

$$\xi^k(\mu(a_h^k)) = \begin{cases} \min\{\vec{t}_{i_h^k}(\rho_{i_h^k}(t_h^k)), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \rho_{i_h^k}(t_h^k) > 1, \\ \min\{\rho_{i_h^k}(t_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.4}$$

$$\psi^k(\mu(a_h^k)) = \max\{\rho_{i_h^k}(t_h^k) + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}. \tag{EC.5}$$

Based on the mapping functions $\rho(.)$, $\sigma(.)$ and with the fact that $e^k + \phi^k(o^k, i_h^k) \leq t_h^k \leq l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)$, $h = 1, ..., \eta^k$, we have that:

(i) for each a_h^k, $h = 1, ..., \eta^k - 1$, $\xi^k(\mu(a_h^k)) \geq t_h^k$;

(ii) for each a_h^k, $h = 2, ..., \eta^k$, $\psi^k(\mu(a_h^k)) \leq t_h^k + \tau_{i_h^k i_{h+1}^k}$.

Thus, we can show that:

(i) $t_1^k - e^k \leq \xi^k(\mu(a_1^k)) - e^k$;

(ii) $l^k - (t_{\eta^k}^k + \tau_{i_{\eta^k}^k d^k}) \leq l^k - \psi^k(\mu(a_{\eta^k}^k))$;

(iii) $t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}) \leq \xi^k(\mu(a_h^k)) - \psi^k(\mu(a_{h-1}^k))$, $h = 2, ..., \eta^k$;

(iv) For each $i \notin P^k$, we have $w_i^k \leq 0$ for constraint (EC.3), hence $w_i^k = 0$.

Hence, $w = \overline{w}$ is proved to be a feasible solution for inequalities (EC.3). Solution $(\bar{x}, \bar{y}, \bar{w})$ is then proved to correspond to a feasible solution (x, y, w) of SND-HC-R($\mathcal{D}_\mathcal{T}$) of the same total fixed, flow and holding cost. Thus, SND-HC-R($\mathcal{D}_\mathcal{T}$) is a valid relaxation of SND-HC($\mathcal{D}^\dagger$).

EC.1.3. Proof of Theorem 2

Let $(\overline{x}, \overline{y}, \overline{w})$ be a feasible solution of formulation SND-HC-R($\overline{\mathcal{D}}_\mathcal{T}$) of cost $\overline{z}$. We show that to solution $(\overline{x}, \overline{y}, \overline{w})$ corresponds a feasible, but not necessarily optimal, solution (x, y, w) of SND-HC-R($\mathcal{D}_\mathcal{T}$), of cost z such that $\overline{z} = z$.

Let $\overline{\mathcal{A}} = \{((i,t),(j,t')) \in \overline{\mathcal{A}}_\mathcal{T} : \overline{y}_{ij}^{tt'} > 0\}$ be the set of arcs traversed by solution $(\overline{x}, \overline{y}, \overline{w})$.

Consider an arc $((i,t),(j,t')) \in \overline{\mathcal{A}}$ such that all arcs of the form $((i,t),(j,t''))$ belong to $\mathcal{A}_\mathcal{T}$ with $t'' < t'$. If no such arc $((i,t),(j,t'))$ exists, solution vectors $x = \overline{x}$ and $y = \overline{y}$ are clearly feasible for constraints (5), (6), (8), and (9). If such arc $((i,t),(j,t'))$ exists, since networks $\overline{\mathcal{D}}_\mathcal{T}$ and $\mathcal{D}_\mathcal{T}$ are defined on the same node set $\mathcal{N}_\mathcal{T}$, a path from (j,t'') and (j,t') exists in $\mathcal{D}_\mathcal{T}$.

Initialize $x = 0$ and $y = 0$ and define $x_{ij}^{k\bar{t}\bar{t}} = \overline{x}_{ij}^{k\bar{t}\bar{t}}$, $k \in \mathcal{K}$, and $y_{ij}^{\bar{t}\bar{t}} = \overline{y}_{ij}^{\bar{t}\bar{t}}$ for all arcs in $((i,t),(j,\bar{t})) \in (\mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T}) \cap (\overline{\mathcal{H}}_\mathcal{T} \cup \overline{\mathcal{A}}_\mathcal{T})$. We can adapt the solution $(\overline{x}, \overline{y}, \overline{w})$ with regard to the arc $((i,t),(j,t'))$ to the solution (x, y, w) concerning the arc $((i,t),(j,t''))$, with the addition of the holding arcs

joining (j,t'') to $(j,t'))$, by setting $y_{ij}^{tt''} = \overline{y}_{ij}^{tt'}$ and $x_{ij}^{ktt''} = x_{jj}^{kt''t'} = \overline{x}_{ij}^{ktt'}$. The resulting (x,y,w) solution is also feasible for constraints (5), (6), (8), and (9), and the process can be repeated for every arc $((i,t),(j,t')) \in \overline{\mathcal{A}}_{\mathcal{T}}$ with $t'' < t'$ for all $((i,t),(j,t'')) \in \mathcal{A}_{\mathcal{T}}$. For each commodity $k \in \mathcal{K}$, let $\overline{P}^k = (\overline{a}_1^k, \ldots, \overline{a}_{\eta^k}^k)$, $\overline{a}_h^k \in \overline{\mathcal{A}}$, $h = 1, \ldots, \eta^k$, be the path induced by solution $(\bar{x}, \bar{y})$ with $\overline{a}_h^k = ((i_h^k, \bar{t}_h^k), (i_{h+1}^k, \overline{\pi}_{h+1}^k))$, $h = 1, \ldots, \eta^k$, where $\bar{t}_h^k$ is the departure time at node i_h^k and $\overline{\pi}_h^k$ is the corresponding arrival time. Due to the definition of solution vectors (x,y) based on solution vectors $(\overline{x}, \overline{y})$, in graph $\mathcal{D}_{\mathcal{T}}$ for commodity k we have a path $P^k = (a_1^k, \ldots, a_{\eta^k}^k)$, $a_h^k \in \mathcal{A}_{\mathcal{T}}$, with departure time $t_h^k = \bar{t}_h^k$, $h = 1, \ldots, \eta^k$, and arrival times $\pi_h^k \leq \overline{\pi}_h^k$, $h = 2, \ldots, \eta^k + 1$.

We now show that solution vector $w = \overline{w}$ satisfies constraints (19)-(20) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), thus showing that (x,y,w) is a feasible SND-HC-R($\mathcal{D}_{\mathcal{T}}$) solution having the same cost of solution $(\overline{x}, \overline{y}, \overline{w})$. First, for each $k \in \mathcal{K}$, $T(\overline{P}^k) = T(P^k)$, hence equations (20) are satisfied. Then, we have that solution vector $\overline{w}$ is defined as:

$$\overline{w}_i^k \leq \begin{cases} \xi^k(\overline{a}_1^k) - e^k, & i = o^k, \\ l^k - \psi^k(\overline{a}_{\eta^k}^k), & i = d^k, \\ \xi^k(\overline{a}_h^k) - \psi^k(\overline{a}_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\ i \in \mathcal{N}, \tag{EC.6}$$

where for each $\overline{a}_h^k$, $h = 1, ..., \eta^k$,

$$\xi^k(\overline{a}_h^k) = \begin{cases} \min\{\vec{t}_{i_h^k}(\bar{t}_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(\bar{t}_h^k) - \bar{t}_h^k > 1\ , \\ \min\{\bar{t}_h^k, l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.7}$$

$$\psi^k(\overline{a}_h^k) = \max\{\bar{t}_h^k + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}. \tag{EC.8}$$

and for inequalities (19) we have

$$w_i^k \leq \begin{cases} \xi^k(a_1^k) - e^k, & i = o^k, \\ l^k - \psi^k(a_{\eta^k}^k), & i = d^k, \\ \xi^k(a_h^k) - \psi^k(a_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\ i \in \mathcal{N}, \tag{EC.9}$$

where for each a_h^k, $h = 1, ..., \eta^k$,

$$\xi^k(a_h^k) = \begin{cases} \min\{\vec{t}_{i_h^k}(t_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(t_h^k) - t_h^k > 1, \\ \min\{t_h^k, l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.10}$$

$$\psi^k(a_h^k) = \max\{t_h^k + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}. \tag{EC.11}$$

Since networks $\overline{\mathcal{D}}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ are defined on the same node set $\mathcal{N}_{\mathcal{T}}$ and $t_h^k = \bar{t}_h^k, h = 1, \ldots, \eta^k$, we have that $\xi^k(a_h^k) = \xi^k(\overline{a}_h^k)$ and $\psi^k(a_h^k) = \psi^k(\overline{a}_h^k)$ for all $h = 1, \ldots, \eta^k$. Constraints (EC.9) thus have the same right-hand side as constraints (EC.6). Hence, $w = \overline{w}$ is proved to be a feasible solution for inequalities (19).

Theorem 2 is thus proved.□

EC.1.4. Proof of Lemma 1

Due to predicate (i) of the lemma, we have that $\psi^k(a)=\bar{t}$ for all $a=((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}$ with $x_{ij}^{kt\bar{t}}=1$ for some $k\in\mathcal{K}$. It follows from the observation that with $(o^k,e^k)\in\mathcal{N}_{\mathcal{T}}$ due to the property 1 of the partially time-expanded network and with $\bar{t}=t+\tau_{ij}$ for each $a=((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}$ with $x_{ij}^{kt\bar{t}}=1$, it must be that $t\geq e^k+\phi^k(o^k,i)$ and thus $\psi^k(a)=\max\{t+\tau_{ij},e^k+\phi^k(o^k,i)+\tau_{ij}\}=t+\tau_{ij}=\bar{t}$ for each $a=((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}$ with $x_{ij}^{kt\bar{t}}=1$. The flat solution $\mathcal{S}$ associated with solution (x,y,w) must be implementable because t and $\bar{t}$ associated with $x_{ij}^{kt\bar{t}}=1$ are feasible arrival and departure times to commodity k in solution $\mathcal{S}$. Each θ_i^k is therefore the corresponding holding time for commodity k at terminal i. This implies that by predicate (ii) of the lemma, that $w_i^k=\theta_i^k$ for all $i\in\mathcal{N}$, $k\in\mathcal{K}$, it must be that (x,y,w) is a feasible solution to the CTSNDP-HC. Since (x,y,w) is an optimal solution of relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), (x,y,w) is therefore proved to be an optimal CTSNDP-HC solution of cost LB. Note that Algorithm 2 computes $UB=\hat{z}_{fc}+\hat{z}_w$ and we have that $LB=z_{fc}(\mathcal{S})+z_w(LB)$. Because the flat solution $\mathcal{S}$ associated with solution (x,y,w) is implementable, we have that $\hat{z}_{fc}=z_{fc}(\mathcal{S})$. Moreover, since Algorithm 2 solves model $IM(\mathcal{S})$ minimizing the total holding cost associated with the flat solution, we also have that $\hat{z}_w=z_w(LB)$, hence $LB=UB$.□

EC.2. Additional details about the computational experiments

EC.2.1. Implementation details

It is worth noting that DDD-based methods rely on two (relative) optimality tolerances: (i) the optimality tolerance used by the DDD algorithm (see parameter *optimality tolerance* of the exact algorithm described in §4), and (ii) the optimality tolerance used by the MIP solver. For the sake of the notation, we denote by tol_{DDD} and tol_{MIP} the two optimality tolerances, respectively. For all considered instances in the two sets of experiments in Section 5, we used $tol_{DDD}=0.01$ for both EXM and EXM-0, $tol_{MIP}=0.01$ for EXM-0. For algorithm EXM, based on our preliminary experiments and as reported by Marshall et al. (2021), we also found to be computationally convenient to dynamically change parameter tol_{MIP}, that starts with 0.04 and is computed as $\max\{gap\times 0.25, 0.098\}$ for each later iteration, where gap is the final gap of the previous iteration. Also, in these experiments, a time limit of two hours is conducted for both EXM and EXM-0. The time limit is imposed over all the iterations, meaning that at each iteration the time limit imposed on the MIP solver is the remaining time. Given tolerance tol_{MIP} applied to the MIP solver and in order to compute safe lower bounds, the lower bound value LB is set equal to the best known bound on the optimal objective given by the Gurobi solver at termination through parameter `ObjBound`.

The algorithms EXM and EXM-0 were implemented in Java language, and Gurobi (v.8.1.1) (Gurobi Optimization 2021) was used as the LP solver to solve model $IM(\mathcal{S})$, and as the MIP solver to solve relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$). The Gurobi function `Model.computeIIS()` was used to

compute IISs in Algorithm 2. The experiments were performed on an Intel(R) Core(TM) i7-8700 (3.20 GHz) Desktop PC equipped with 64 GB RAM running under Windows 10 64-bit operating system.

EC.2.2. Generating instances based on CTSNDP benchmark instances

Table EC.1 gives the details of the 31 classes of networks $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ considered by Crainic et al. (2001). In the table, the column "Cost ratio" computed as $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\frac{f_a}{c_a u_a}$ measures the ratio between fixed and variable costs, "Cap ratio" computed as $\sum_{k\in\mathcal{K}} q^k / \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} u_a$ indicates whether the arcs are loosely or tightly capacitated, and "Avg length" computed as $\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\phi^k(o^k, d^k)$is the average length of the least total travel time paths. For each of the 31 classes of networks reported in the table, Boland et al. (2019) generated 18 CTSNDP timed instances by first calculating the travel times for each arc and then by generating the time windows for each commodity by randomly sampling from a normal distribution.

Table EC.1 Instances from Crainic et al. (2001)

Class	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$	Cost ratio	Cap ratio	Avg length
c33	20	228	39	0.02	5.8	2407.9
c35	20	230	40	0.02	16.0	767.9
c36	20	230	40	0.08	16.0	3705.8
c37	20	228	200	0.51	16.0	1871.4
c38	20	230	200	0.97	16.0	4381.0
c39	20	229	200	0.47	20.0	1691.3
c40	20	228	200	0.94	22.0	3522.1
c41	20	288	40	0.02	8.0	1622.0
c42	20	294	40	0.08	10.0	5675.8
c43	20	294	40	0.02	16.0	776.5
c44	20	294	40	0.08	16.0	3517.9
c45	20	294	200	0.48	25.0	1124.2
c46	20	292	200	1.01	25.0	2632.0
c47	20	291	200	0.46	28.0	996.6
c48	20	291	200	0.95	28.0	2271.6
c49	30	518	100	0.10	20.0	341.1
c50	30	516	100	0.51	20.0	1586.5
c51	30	519	100	0.09	29.9	206.6
c52	30	517	100	0.49	29.9	1161.5
c53	30	520	400	0.18	40.0	612.1
c54	30	520	400	0.36	40.0	1061.8
c55	30	516	400	0.18	49.9	479.4
c56	30	518	400	0.35	49.9	966.9
c57	30	680	100	0.09	20.0	307.6
c58	30	680	100	0.20	20.0	592.8
c59	30	687	100	0.10	29.9	187.1
c60	30	686	100	0.20	29.9	394.7
c61	30	685	400	0.19	40.0	503.8
c62	30	679	400	0.36	40.0	1056.5
c63	30	678	400	0.18	49.9	381.4
c64	30	683	400	0.34	49.9	780.0

For each of the 558 CTSNDP instances, we generated a CTSNDP-HC instance by setting the per-unit-of-demand-and-time holding cost h_i^k for each commodity $k \in \mathcal{K}$ at each terminal $i \in \mathcal{N}$ as follows. We set the holding cost using the scheme proposed by Lai et al. (2022), in which fixed, flow, and holding costs were defined for the LTL shipment case. We observe that the average per-unit-of-demand-and-time transportation cost for the 558 CTSNDP instances, calculated as $\frac{\sum_{a\in\mathcal{A}}((c_a+f_a/u_a)/\tau_a)}{|\mathcal{A}|}$, is nearly 0.02. The value obtained is comparable with the transportation cost per hundred weights per unit of Euclidean distance, nearly 0.03, used in Lai et al. (2022). Note that Lai et al. (2022) utilized a discount cost function for the flow cost expressed in dollars per hundred weights. The transportation cost per hundred weight per unit of Euclidean distance can be approximated as $\frac{\sum_{a\in\mathcal{A}}((c_a^{max}+(100\,f_a)/u_a)/d_a)}{|\mathcal{A}|}$, where c_a^{max} represents the maximum unit charge and d_a represents the Euclidean distance of the arc. In Lai et al. (2022), the holding cost per weight per time period is randomly sampled in the interval [0.025, 0.1], with a holding cost per hundred weights per unit of Euclidean distance ranging from 0.0025 to 0.02. In addition, the arc travel times are computed as a function of the Euclidean distances associated with the arcs. Therefore, we set the per-unit-of-demand-and-time holding cost as follows. For each commodity $k \in \mathcal{K}$, we randomly sample value h^k from the interval [0.0025, 0.02], then for each terminal $i \in \mathcal{N}$, the per-unit-of-demand-and-time holding cost h_i^k is randomly sampled from $[0.8 \times h^k, 1.2 \times h^k]$. Since, in practice, a commodity does not incur any holding costs after it reaches the destination, for each commodity $k \in \mathcal{K}$ we set $h_{d^k}^k = 0$. We, therefore, generated 558 CTSNDP-HC instances.

EC.2.3. Newly generated CTSNDP-HC instances

The procedure used to generate the new instances follows two main steps.

(1) **Varying the connectivity level**. For each instance of Table EC.1 and the corresponding network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, we first derive a timed instance using the method proposed by Boland et al. (2019), i.e., we compute the travel times τ_{ij}. We then generate the per-unit-of-demand-and-time cost h_i^k using the method explained in §EC.2.2. Let Γ be the time of the path in $\mathcal{D}$ having maximum time among the least time $o^k - d^k$ paths associated with the vertices o^k, d^k, $\forall k \in \mathcal{K}$. Then, we reduce the number of arcs of the network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ by an arc reduction procedure that at each iteration performs the following steps:

(i) Randomly select an arc a by means of a uniform distribution from the set of arcs $\mathcal{A}$.

(ii) Check the connectivity of the network $(\mathcal{N}, \mathcal{A} \setminus \{a\})$, i.e., check if for every pair of vertices o^k, d^k, $k \in \mathcal{K}$, there exists a path connecting o^k to d^k.

(iii) If the graph is connected, set $\mathcal{A} = \mathcal{A} \setminus \{a\}$ and begin a new iteration.

If the removal of an arc a results in a disconnected network, a new arc is randomly selected and the procedure terminates after $\frac{1}{10}x$ unsuccessful removal attempts, where x is the initial

number of arcs. After termination, each remaining arc in the resulting graph $\mathcal{D}$ is in turn selected and tested for removal.

Let NR be the total number of arcs removed. We consider four final networks corresponding to the networks obtained after the removal of $\lfloor xNR \rfloor$ arcs where $x \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$, denoted as $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$, respectively.

(2) **Varying the flexibility level**. Given a network $\mathcal{D}_x$, $x = 1, 2, 3, 4$, let $\underline{\tau}_{ij}$, $i, j \in \mathcal{N}$, $i \neq j$ be the length of the least total travel time path from i to j, and let $\mathcal{B} = \{(i, j) : \underline{\tau}_{ij} \leq \Gamma\}$.

If, for a commodity $k \in \mathcal{K}$, we have $\underline{\tau}_{o^k d^k} > \Gamma$, we assign to the commodity new origin and destination nodes by randomly sampling with a uniform distribution a new pair from set $\mathcal{B}$, and we recompute the new value $\underline{\tau}_{o^k d^k}$.

We then generate earliest and latest times also based on the method proposed by Boland et al. (2019) as follows:

(a) We compute the average length computed as $l_{avg} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \phi^k(o^k, d^k)$.

(b) For generating values e^k, we create a normal distribution with mean l_{avg} and standard deviation $\frac{1}{6} l_{avg}$.

(c) For generating values l^k, we create three normal distributions (denoted as A, B and C, respectively) from which we drawn values l^k, all of which are defined by a standard deviation $\frac{1}{6}\mu$ but where we consider the values for the mean μ, $\frac{1}{2} l_{avg}$, l_{avg} and $\frac{3}{2} l_{avg}$. A value l^k is set equal to $e^k + \phi^k(o^k, d^k) + \alpha_k$ where α_k is the value drawn from a distribution.

Based on the above two steps, for each of the instances in Table EC.1, we generate four different networks $\mathcal{D}_x$, $x = 1, 2, 3, 4$, and three instances based on the three different distributions for values e^k and l^k. These steps are repeated three times to finally obtain a total of $3 \times (31 \times 4 \times 3) = 1116$ instances.

EC.2.4. Preliminary results for EXM-0

Here we summarize the results obtained by the baseline algorithm EXM-0 in solving the 558 CTSNDP instances, and we then compare its performance with the results of BHMS17 and MBSH21 reported in Marshall et al. (2021). A time limit of one hour was imposed to EXM-0, as done for both BHMS17 and MBSH21.

In the comparison reported by Marshall et al. (2021), for method BHMS17, tol_{DDD} and tol_{MIP} were set equal to 0.01 and 0.0001 (i.e., the CPLEX default setting), respectively. For method MBSH21, tol_{DDD} was also set equal to 0.01, whereas tolerance tol_{MIP} was dynamically changed during the different iterations. More precisely, the initial tolerance is set equal to 0.04, and then for each iteration tol_{MIP} is computed as $\max\{gap \times 0.25, tol_{DDD} \times 0.98\}$, where gap is the final gap of the previous iteration. Note that EXM-0 applies the same adaptive optimality tolerance.

Table EC.2 Summary results on the CTSNDP instances

Group	Algorithm	$\%UB$	*time*	*iter*	$\%opt$
HC/LF	BHMS17	0.08	1391.1	5.3	77.1
183	MBSH21	0.12	677.8	14.8	85.8
	EXM-0	1.00	236.3	10.0	96.7
HC/HF	BHMS17	0.56	1966.7	6.0	53.7
177	MBSH21	0.84	1693.8	17.5	56.5
	EXM-0	3.18	1555.5	11.0	65.0
LC/LF	BHMS17	0.00	28.6	3.7	100.0
94	MBSH21	0.00	0.6	6.5	100.0
	EXM-0	0.70	0.5	3.5	100.0
LC/HF	BHMS17	0.00	1.5	2.5	100.0
104	MBSH21	0.00	0.1	3.2	100.0
	EXM-0	0.55	0.1	1.6	100.0

For the set of CTSNDP instances, our comparison is based on the results reported by Marshall et al. (2021) which were obtained on a single core machine using CPLEX 12.6 as the MIP solver (for both BHMS17 and MBSH21, with no specific machine type being reported). Because the computational environment of BHMS17 and MBSH21 was different from that of our algorithms, a direct comparison is therefore not possible. However, in what follows we give a clear overall picture of the relative performance, especially when the total number of instances solved to proven optimality is compared.

Table EC.2 gives the comparison of the three algorithms. For each group of instances, the table shows the number of instances in the group and, for each algorithm, the average percentage deviation of the final upper bound UB computed with respect to the final lower bound LB ("$\%UB$"), i.e., $100.0 \times \frac{UB-LB}{UB}$, the average computing time in seconds ("*time*"), the average number of iterations ("*iter*"), and the percentage of the instances solved to optimality ("$\%opt$") (within the given optimality tolerance). The average values over all instances are computed. For each method, the upper bound UB corresponds to the cost of the best solution found by the method over the different algorithm iterations and the lower bound LB is computed as the maximum among the lower bounds computed at the different iterations.

The table shows that our implementation of the algorithm of Boland et al. (2017) compares well with both algorithms BHMS17 and MBSH21, and also shows similar performance on the different groups of instances. We note that the computational environment of EXM-0 is different from the one used by the other methods. However, the comparison over the total number of instances solved to proven optimality gives a clear global picture of the relative performance.

In groups **LC/LF** and **LC/HF**, EXM-0 was capable of solving to optimality all the instances within the imposed optimality tolerance. In these groups, with respect to BHMS17 and MBSH21,

EXM-0 shows higher percentage deviations of the final upper bound UB. This can be due to the fact that different MIP solvers are used, and that EXM-0 computes a safe lower bound based on the best-known bound given by the Gurobi MIP solver. Based on the results of Table EC.2, we adopted algorithm EXM-0 as a baseline algorithm for comparison purposes with EXM.

EC.2.5. Additional results on the effect of the algorithm components

The heuristic method described by Algorithm 2 removes at each iteration the infeasible consolidation constraint (12d) having the minimum increase of the fixed cost. To illustrate the effectiveness of our choice, below, we compare alternative methods to handle infeasible consolidation constraints (12d). More specifically, we consider alternative upper bounds, namely $UBM1$ and $UBM2$, where the infeasible constraint during Algorithm 2 is removed based on the following rules: (i) $UBM1$: the infeasible constraint with the earliest departure time. (ii) $UBM2$: the infeasible constraint with the tightest time flexibility on the selected consolidation arc (i,j), i.e., $(l^k - \phi^k(j,d^k) - \tau_{ij}) - (e^k + \phi^k(o^k,i))$. In addition, we also consider upper bound $UB1$ described in Section 5.1.2. Table EC.3 gives the comparison, where $\%UBx = 100.0 \times \frac{UBx - UB}{UBx}$, $x \in \{M1, M2, 1\}$, is the percentage deviations of upper bound UBx with respect to the upper bound UB computed by Algorithm 2. The comparison concerns the different upper bounds $UBM1$, $UBM2$, and UB computed at the first iteration of algorithm EXM and upper bound $UB1$ computed at the first iteration of algorithm EXM-0. The table shows that the different upper bounds show similar results in the LC instances. In contrast, the upper bound UB is significantly better than the other upper bounds in the HC instances, thus attesting to the effectiveness of our choice.

Table EC.3 Computational results for EXM variants with different refinement strategies

	$\%UBM1$	$\%UBM2$	$\%UB1$
HC/LF	0.9	1.1	0.9
HC/HF	2.5	2.9	2.0
LC/LF	0.1	0.1	0.1
LC/HF	0.1	0.1	0.1

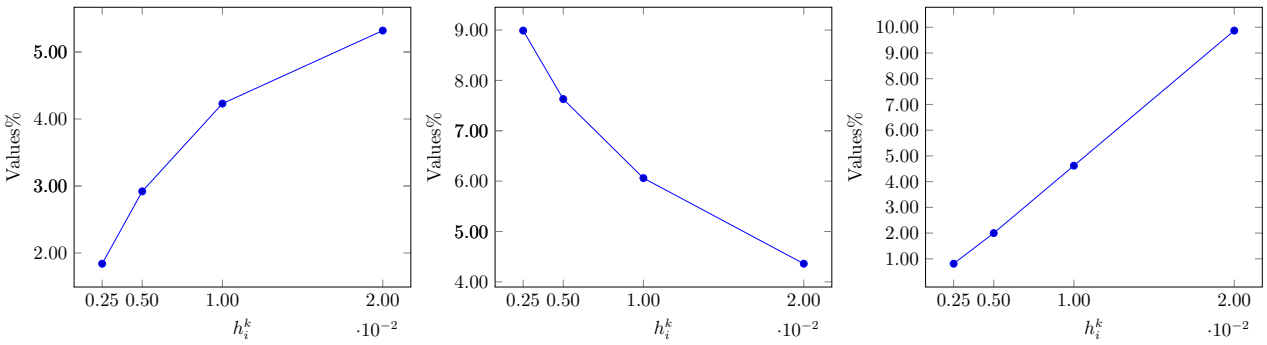
To attest the effectiveness of the refinement strategies, we executed algorithm EXM with refining different sets of short-arcs. More preciously, we conducted algorithm EXM without refining the short-arcs identified by Algorithm 2 in refinement strategy 1, referred to as EXM_R1 and by selecting at most $|\mathcal{K}|/10$ short-arcs for the additional set in refinement strategy 1, referred to as EXM_R2. In the case of EXM_R2, we select fewer arcs than the limit of $|\mathcal{K}|/5$ arcs used by EXM. The results are shown in Table EC.4 using the same notation introduced in Table 2. The results show that the algorithm EXM benefits from a slightly aggressive refinement strategy 1 that refines all short-arcs identified by Algorithm 2 and refines more short-arcs for the additional set. Indeed, EXM shows a better optimality rate and optimality gap for unsolved instances compared to EXM_R1 and EXM_R2.

Table EC.4 Computational results for EXM variants with different refinement strategies

Group	Algorithm	%UB			time	iter	%opt
		min	*max*	*avg*			
HC/LF	EXM_R1	1.17	1.9	1.7	277.2	4.4	98.4
	EXM_R2	1.10	2.0	1.6	372.2	6.1	97.8
	EXM	1.08	1.9	1.6	279.3	4.4	98.4
HC/HF	EXM_R1	1.01	8.9	3.4	3107.7	8.7	60.5
	EXM_R2	1.00	9.9	3.4	3118.0	8.9	62.7
	EXM	1.01	6.1	2.9	2902.7	6.2	65.5
LC/LF	EXM_R1	-	-	-	0.7	1.8	100.0
	EXM_R2	-	-	-	0.6	1.7	100.0
	EXM	-	-	-	0.7	1.8	100.0
LC/HF	EXM_R1	-	-	-	0.1	2.3	100.0
	EXM_R2	-	-	-	0.2	3.3	100.0
	EXM	-	-	-	0.2	2.3	100.0

EC.2.6. Sensitive analysis on the holding costs

To analyze the effect of varying the per-unit-of-demand-and-time cost h_i^k on the CTSNDP-HC we compare the results obtained by EXM for the case with different per-unit-of-demand-and-time cost h_i^k, i.e., $h_i^k \in \{0.0025, 0.05, 0.01, 0.02\}$. For the experiments, we considered a restricted set of instances composed of the instances of networks $\mathcal{D}_2$ and $\mathcal{D}_3$ under the three different distributions A, B, and C.



(a) $\%hc$ (b) $\%ht$ (c) $\%UB1_{avg}$

Figure EC.1 Sensitive analysis on the per-unit-of-demand-and-time cost h_i^k

Figure EC.1 summarizes the results obtained. For each per-unit-of-demand-and-time cost h_i^k (x axis), the figure shows the following average percentage values: (i) $\%hc$, the holding cost over the total solution cost, (ii) $\%ht$, the holding time over the total transit time, and (iii) $\%UB1_{avg}$, the average percentage deviation of upper bound $UB1$ with respect to upper bound UB. The percentage values were computed over all instances solved to optimality with all the considered unit holding values. The figure shows that when unit holding cost increases, both $\%UB1_{avg}$ and $\%hc$

increase. This implies that although a higher per-unit-of-demand-and-time holding cost induces a larger percentage of the holding cost, it induces a more significant cost saving achieved by UB over $UB1$. From the figure, we can also observe that when unit holding cost increases, $\%ht$ decreases. This implies that a higher per-unit-of-demand-and-time holding cost also induces a significant reduction in holding time. Hence, as per-unit-of-demand-and-time holding cost increases, it is more beneficial to take into account holding costs for solving the CTSNDP-HC. We notice that, with a per-unit-of-demand-and-time holding cost of 0.0025, the percentage deviation of upper bound $UB1$ is less than 1%, thus on average, a per-unit-of-demand-and-time holding cost around or below 0.0025 will make a negligible difference between UB and $UB1$ on the considered instances. However, the final holding cost also depends on other cost ratios, time flexibility, and network connectivity. Thus, the exact cut-off value for per-unit-of-demand-and-time holding costs differs for different instances.