Further Details on Computational Experiments

Shengnan Shu
Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, shengnan.shu@connect.polyu.hk

Zhou Xu
Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, lgtzx@polyu.edu.hk

Jin Qi
Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology, jinqi@ust.hk

1. Instance Generation for Computational Experiments

For our computational experiments, we generated instances of the robust CTSNDP based on the seven instance classes (named R4-R10) of the fixed-charge capacitated multi-commodity network design (CMND) problem available in the literature Ghamlouche et al. (2003). These classes of CMND instances have been utilized in previous studies to generate test instances for various stochastic capacitated fixed charge network design problems (see, for example, Crainic et al. 2011, Sarayloo et al. 2021a,b). Their sizes are comparable to those in previous studies on SNDP under uncertainties (Wang and Qi 2020, Lanza et al. 2021).

The sizes of the node set $\mathcal{N}$, arc set $\mathcal{A}$, and commodity set $\mathcal{K}$ vary from 10 to 20, from 60 to 120, and from 10 to 50, respectively, among instances belonging to different classes. These instances are referred to as "untimed" instances, as they do not have any temporal attributes such as travel times of arcs, or earliest available times and due times of commodities. Moreover, each of these instance classes consists of five networks with different "ratio" indices valued from 1, 3, 5, 7, to 9. These index values indicate different combinations of ratios of fixed cost to variable cost and total demand to total network capacity, with values of (0.01, 1), (0.1, 1), (0.05, 2), (0.01, 8), and (0.1, 8), respectively.

To obtain "timed" instances for each of the 7 classes of the CMND problem, we followed an approach similar to that presented in Boland et al. (2017) to generate fixed costs and time attributes of the CTSNDP. First, for each arc $(i, j) \in \mathcal{A}$, we set the nominal value of travel time (in minutes) $\overline{\tau}_{ij}$ to be proportional to its fixed cost f_{ij} by setting $\overline{\tau}_{ij} = f_{ij}/0.55$, as in Boland et al. (2017). This is based on the same premise that f_{ij} represents the transportation cost for carriers that spend 0.55 cents per mile, and that their trucks travel at 60 miles per hour.

Next, for each commodity $k \in \mathcal{K}$, we followed a normal distribution to randomly generate the available time e^k. Let $\mathscr{L}_k$ denote the length of the shortest-time path from origin o^k to destination d^k for commodity k in the flat network under the nominal travel times $\overline{\boldsymbol{\tau}}$. We then set the due time l^k of each commodity $k \in \mathcal{K}$ by $l^k = e^k + \mathscr{L}_k + \mathscr{F}_k$. Here, the parameter $\mathscr{F}_k \geq 0$ represents the time flexibility for the delivery of commodity k, which we also set randomly using a normal distribution. We used the same normal distribution to generate the available times e^k for all instances, but used two different normal distributions to generate $\mathscr{F}_k$ for instances of high and low time flexibility, respectively. Consequently, we had two combinations of normal distributions to generate commodities' available times and time flexibility. The detailed settings of these normal distributions are described in Table 1.1, where $\mathscr{L}$ denotes the average of $\mathscr{L}_k$ over all $k \in \mathcal{K}$.

Table 1.1 Detail setting of the normal distributions used for generating "timed" instances.

Normal Distribution	Mean(μ)	Standard Deviation(σ)
For generating e_k	$\mathscr{L}$	$\frac{1}{6}\mathscr{L}$
For generating $\mathscr{F}_k$	$\frac{1}{2}\mathscr{L}$ $\frac{1}{4}\mathscr{L}$	$\frac{1}{6} \cdot \frac{1}{2}\mathscr{L}$ $\frac{1}{6} \cdot \frac{1}{4}\mathscr{L}$

For each "timed" instance obtained, we then generated unit in-storage holding costs and unit delay penalties for the commodities. We set the per-unit-of-demand-and-time in-storage holding cost h^k for each commodity $k \in \mathcal{K}$ to be proportional to its cheapest per-unit-of-time per-unit-of-flow cost, i.e., $h^k = 0.5 \min_{a \in \mathcal{A}} \{(c_a^k + f_a/u_a)/\overline{\tau}_a\}$ where $\overline{\tau}_a$ is the nominal value of the travel time generated. Inspired by Lanza et al. (2021), for each commodity $k \in \mathcal{K}$, we set its penalty g^k per unit of time for the delay to be twice the most expensive per-unit-of-time transportation cost for it to pass through an arc, i.e., $g^k = 2 \cdot \max_{a \in \mathcal{A}} \{(c_a^k \cdot q^k + f_a \lceil q^k/u_a \rceil)/\overline{\tau}_a\}$.

Moreover, to characterize travel time uncertainty, we generated the maximum deviation $\hat{\tau}_{ij}$ of the travel time for each arc $(i,j) \in \mathcal{A}$ by setting $\hat{\tau}_{ij} = \hat{\mu}_{ij}\overline{\tau}_{ij}$. Here, $\overline{\tau}_{ij}$ is the nominal value of the travel time generated, and $\hat{\mu}_{ij}$ is a coefficient randomly selected from 0.1 to 0.5.

For each network in each problem class, we randomly generated 3 instances for each combination of the distributions for commodities' available time and time flexibility. As a result, we obtained $5 \times 2 \times 3 = 30$ test instances for each of the 7 instance classes, and thus obtained $7 \times 30 = 210$ test instances in total.

In our second set of experiments, under different values of Γ and $\mathcal{Z}$, we need to evaluate the worst-case and average total costs of solutions obtained from models RO, RS and SP over randomly generated scenarios for instances in class R7. For each of the 30 instances in class R7, we generated 200 scenarios at random to create the set Π for model SP and model MM. To generate each scenario $\boldsymbol{\delta}$, we drew each of the realization of δ_{ijr} for $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$ uniformly from the set $\{-1, -(\hat{\tau}_{ij}-1)/\hat{\tau}_{ij}, -(\hat{\tau}_{ij}-2)/\hat{\tau}_{ij}, \cdots, -1/\hat{\tau}_{ij}, 0, 1/\hat{\tau}_{ij}, \cdots, (\hat{\tau}_{ij}-1)/\hat{\tau}_{ij}, 1\}$.

2. Additional Computational Results

In this section, additional computational results are reported.

2.1. Details on Performance of RO-C&CG and RS-C&CG

Table 2.1 below presents the computational performances of RO-C&CG and RS-C&CG across instances with different cost and time attributes. As mentioned in Section 1, each CTSNDP instance origins from an untimed instance with a "ratio" index. The "ratio" index is valued from 1, 3, 5, 7, to 9, indicating combinations of ratios of fixed cost to variable cost ratio and total demand to total network capacity as (0.01, 1), (0.1, 1), (0.05, 2), (0.01, 8), and (0.1, 8), respectively. Moreover, each instance is associated with time flexibility parameters $\mathscr{F}_k$ for $k \in \mathcal{K}$, which follow a normal distribution with its mean equal to $\mathscr{L}/2$ or $\mathscr{L}/4$.

Accordingly, we regrouped the 210 CTSNDP instances into ten groups, according to their "ratio" indices and their means of time flexibility. In Table 2.1, for each instance group and each C&CG algorithm, we report the percentage of the instances solved to optimality in column opt%, the average optimality gap in column g% (defined as the percentage gap between the best upper and lower bounds found), the average computational time in CPU seconds in column T for C&CG algorithms for RO and RS. Moreover, we present the number of required directed shipping services in column #service, the percentage of services consolidating two or more commodities in column cons%, along with the flow cost and the fixed cost, for the final solution obtained by DO, RO-C&CG, and RS-C&CG.

From the results in Table 2.1 we can obtain the following insights. First, it can be seen that instances with greater time flexibility are more difficult to solve, especially when the ratio of fixed cost to variable cost and the ratio of total demand to total network capacity are both moderately large. Second, compared to the optimal deterministic solutions, robust solutions typically require more directed shipping services but fewer consolidations, thereby incurring higher transportation costs. Third, robust solutions often exhibit both higher average flow costs and fixed costs, resulting in higher total transportation costs. However, this trend does not always hold, since for certain instances, the flow costs or fixed costs of robust solutions may be lower than those of the optimal deterministic solutions.

Table 2.1 Details on Performance of RO-C&CG and RS-C&CG Algorithms

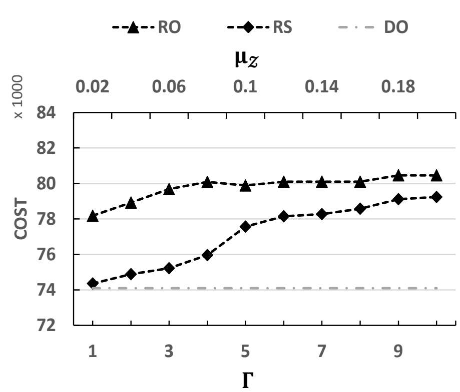
Ratio Index	Time Flexibility	DO				RO-C&CG							RS-C&CG						
		#service	cons%	flow cost	fixed cost	Opt%	g%	T	#service	cons%	flow cost	fixed cost	Opt%	g%	T	#service	cons%	flow cost	fixed cost
1	$\frac{1}{2}\mathscr{L}$	35.9	21.7	129058.3	30891.0	95.2	0.0	2920.4	38.3	19.3	139691.1	30643.6	95.2	0.5	1747.4	37.3	18.5	135181.1	30614.05
	$\frac{1}{4}\mathscr{L}$	39.1	16.4	143799.1	31009.6	100.0	0.0	52.5	41.3	13.4	146870.3	32109.7	100.0	0.0	39.6	41.2	13.3	146527.6	32229.33
3	$\frac{1}{2}\mathscr{L}$	33.2	36.5	153883.0	237284.9	85.7	0.4	6901.3	34.8	32.1	157656.3	248557.0	85.7	8.7	6514.6	34.6	31.5	157865.4	247540.1
	$\frac{1}{4}\mathscr{L}$	38.5	22.7	155527.2	284157.8	100.0	0.0	56.7	40.0	19.8	158075.1	293721.8	100.0	0.0	114.4	40.1	19.9	157175	293983.2
5	$\frac{1}{2}\mathscr{L}$	36.3	26.7	142456.7	151308.8	81.0	0.1	8244.2	38.1	22.8	148480.3	161535.0	81.0	4.8	7191.3	37.9	22.6	144665.1	158994.3
	$\frac{1}{4}\mathscr{L}$	40.3	16.9	149032.9	171943.5	100.0	0.0	52.6	42.2	13.9	153913.7	178427.3	100.0	0.0	154.1	41.9	14.2	152576.6	177666.7
7	$\frac{1}{2}\mathscr{L}$	40.6	12.4	135161.7	75372.4	95.2	0.0	1969.6	42.6	11.1	141932.9	78952.0	85.7	1.7	4776.1	42.1	10.1	137479.2	79312.1
	$\frac{1}{4}\mathscr{L}$	42.2	9.2	144553.3	81116.8	100.0	0.0	83.6	43.8	8.5	148991.5	83608.5	100.0	0.0	13.8	43.6	7.7	146458.5	83101.33
9	$\frac{1}{2}\mathscr{L}$	38.8	21.2	148237.5	505414.1	85.7	0.2	4413.6	41.2	16.9	153465.2	531381.4	90.5	3.6	4247.4	40.6	17.5	152138.4	522552
	$\frac{1}{4}\mathscr{L}$	42.6	12.5	152407.3	571584.3	100.0	0.0	19.2	43.6	10.6	154568.5	583166.0	100.0	0.0	65.0	43.6	9.8	153090.0	585582.5

2.2. Nominal Performance of Robust Solutions

We conducted additional computational experiments across 30 instances of class R7 to evaluate the performance of the robust solutions obtained from models RO and RS under the nominal scenario. For each instance in R7, we used RO-C&CG to solve model RO for each uncertainty budget $\Gamma \in \{1, 2, ..., 10\}$, and used RS-C&CG to solve model RS for each cost target $\mathcal{Z} = \lceil (1 + \mu_z) \cdot \mathcal{Z}_0 \rceil$ with $\mu_z \in \{0.02, 0.04, ..., 0.2\}$, where $\mathcal{Z}_0$ is set as the optimal objective value of model DO. For each robust solution obtained, we computed the total cost of its first-stage solution in the nominal scenario. We then solved model DO to obtain the minimum achievable total cost for the nominal scenario.

The results are presented in Figure 1, where the total cost along the vertical axis is the mean across all instances in R7. It can be seen that robust solutions typically exhibit inferior nominal performance compared to the optimal deterministic solution, since they prioritize the optimization of worst-case performance. However, solutions obtained from model RS exhibit better overall performance in the nominal scenario than those from model RO. This is because the robust optimization approach focuses exclusively on the worst-case total cost within the budget of uncertainty Γ, whereas model RS is solved within a specified acceptable cost loss relative to the nominal optimal solution under the nominal scenario, denoted as $\mathcal{Z} - \mathcal{Z}_0$. Moreover, it can be seen that when μ_z increases, the total nominal cost of the solutions from model RS gradually increases.

Figure 1 Total costs in nominal scenario



In contrast, the total nominal cost of solutions obtained from model RO does not show a strictly monotonically increasing trend as Γ increases. This suggests that the cost target $\mathcal{Z}$ is more effective in controlling the price of robustness for robust solutions in terms of their performance loss in the nominal scenario.

2.3. Results for Large Instances

We followed the method described in Section 1 to generate 150 large CTSNDP instances from five classes (R11-R15) of the fixed-charge capacitated multi-commodity network design (CMND) problem in Ghamlouche et al. (2003), with each class containing 30 instances. For these large CTSPND instances, the number of nodes $|\mathcal{N}|$, the number of arcs $|\mathcal{A}|$, and the number of commodities $|\mathcal{K}|$ are presented in Table 2.2. For each of these large instances, we solved the deterministic model DO using the Gurobi solver within an 8-hour time limit, and set its final objective value as the cost target $\mathcal{Z}_0$ for RS-C&CG. The average optimality gap (g%) and computational time in CPU seconds (T) of model DO are presented in Table 2.2. Moreover, we set the uncertainty budget Γ of model RO and the cost target $\mathcal{Z}$ of model RS to $\lceil 0.05 \cdot |\mathcal{K}| \rceil$ and $\lceil (1+0.05) \cdot \mathcal{Z}_0 \rceil$, respectively.

The computational results are detailed in Table 2.2, including the percentage of instances solved optimally (opt%), the average optimality gap (g%), and the average computational time in CPU seconds (T) for each C&CG algorithm across different instance classes. Additionally, we present the mean and maximum of improvement ratio against the obtained deterministic solution of model DO, with respect to the objective value of the corresponding robust model, in columns under Im%. Specifically, the improvement ratio for each instance is computed by $\frac{C_{DO}-\mathrm{UB}}{C_{DO}}$, where C_{DO} represents the objective value of the corresponding robust model achieved by the obtained deterministic solution and UB represents the best upper bound value found for the corresponding robust model.

Results in Table 2.2 indicate that both model RO and model RS for instances of 100 or more commodities are challenging to solve, highlighting the need for future studies to enhance the

Table 2.2 Computational Performance of RO-C&CG and RS-C&CG Algorithms over Large Instances

Class	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$	DO		RO-C&CG					RS-C&CG				
									Im%					Im%	
				g%	T	opt%	g%	T	mean	max	opt%	g%	T	mean	max
R11	20	120	100	1.9	24435.5	0.0	13.2	28800.0	9.6	24.1	0.0	96.4	28800.0	41.1	77.8
R12	20	120	200	5.0	28800.0	0.0	21.5	28800.0	6.3	20.5	0.0	100.0	28800.0	21.9	46.8
R13	20	220	40	0.0	1.9	100.0	0.0	2398.0	6.3	13.4	100.0	0.0	4562.6	38.4	80.3
R14	20	220	100	0.1	7919.5	0.0	7.5	28800.0	7.7	17.5	0.0	75.7	28800.0	39.2	71.8
R15	20	220	200	3.7	28800.0	0.0	19.1	28800.0	6.6	18.6	0.0	100.0	28800.0	17.1	57.0
Mean				2.1	17991.4	20.0	12.3	23519.6	7.3	24.1	20.0	74.4	23952.5	31.6	80.3

solutions algorithms. However, the columns under Im% indicate that compared to the deterministic solutions obtained from model DO, the robust solutions from our RO-C&CG and RS-C&CG algorithms have significantly improved robust objective values with respect to model RO and model RS. Specifically, the improvements in the worst-case total cost and the worst-case normalized deviation from the prescribed cost target are 7.3% and 31.6% on average, and 24.1% and 80.3% at maximum, respectively. This demonstrates the benefit of the robust models in generating reliable CTSNDP solutions.

References

Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. Operations Research 65, 1303–1321.

Crainic, T.G., Fu, X., Gendreau, M., Rei, W., Wallace, S.W., 2011. Progressive hedging-based metaheuristics for stochastic network design. Networks 58, 114–124.

Ghamlouche, I., Crainic, T.G., Gendreau, M., 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. Operations Research 51, 655–667.

Lanza, G., Crainic, T.G., Rei, W., Ricciardi, N., 2021. Scheduled service network design with quality targets and stochastic travel times. European Journal of Operational Research 288, 30–46.

Sarayloo, F., Crainic, T.G., Rei, W., 2021a. A learning-based matheuristic for stochastic multicommodity network design. INFORMS Journal on Computing 33, 643–656.

Sarayloo, F., Crainic, T.G., Rei, W., 2021b. A reduced cost-based restriction and refinement matheuristic for stochastic network design problem. Journal of Heuristics 27, 325–351.

Wang, Z., Qi, M., 2020. Robust service network design under demand uncertainty. Transportation Science 54, 676–689.