Implementation Details on Benchmark Algorithms

Shengnan Shu
Department of Civil and Environmental Engineering, National University of Singapore, shengnan@nus.edu.sg,

Lubin Fan
Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), lfan561@connect.hkust-gz.edu.cn,

Shuai Jia
Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), shuaijia@hkust-gz.edu.cn,

Qiang Meng
Department of Civil and Environmental Engineering, National University of Singapore, ceemq@nus.edu.sg.

---

1. Parameterized Column-and-Constraint Generation Solution Method

In this section, we introduce how to apply the parameterized C&CG algorithm proposed by Zeng and Wang (2022) to solve model RO, which is formulated as the following two-stage optimization model:

$$
\begin{aligned}
[\mathrm{RO}] \quad & \min \sum_{k\in\mathcal{K}}\sum_{b=1}^{B_k} c_k y_{kb} + \sum_{k\in\mathcal{K}}\sum_{b=1}^{B_k} f_k \sigma_{kb} + Q(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v}) \\
& s.t.\ (\boldsymbol{y}_k,\boldsymbol{\sigma}_k,\boldsymbol{v}_k) \in \mathcal{Q}(\boldsymbol{x},k,B_k,H_k), \quad \forall k\in\mathcal{K} \\
& \boldsymbol{x}\in\mathcal{X}
\end{aligned}
$$

where $Q(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})$ indicates the worst-case second-stage cost and can be calculated by the following max-min optimization model, referred to as model RP$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})$.

$$Q(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v}) = \max_{\tilde{\boldsymbol{R}}\in\mathcal{U}(\boldsymbol{x},\Delta)} \min \sum_{i\in\mathcal{I}} C_i t_i \tag{1.1}$$

$$s.t.\ r_{kb} \geq \tilde{R}_i - M(1-\sum_{p=1}^{H_k} x_{ikbp}), \quad \forall k\in\mathcal{K}, i\in\mathcal{I}_k, b\in\{1,...,B_k\} \tag{1.2}$$

$$u_{kb} \geq \sum_{m=1}^{Z_k} U_m^k \cdot v_{kbpm} - U_k(1-y_{kb}), \quad \forall k\in\mathcal{K}, b\in\{1,...,B_k\}, p=\{1,...,H_k\} \tag{1.3}$$

$$s_{kb} \geq s_{k,b-1} + r_{k,b-1} + u_{kb}, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k\} \tag{1.4}$$

$$s_{kb} \geq A_i \sum_{p=1}^{H_k} x_{ikbp}, \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, b \in \{1,...,B_k\} \tag{1.5}$$

$$t_i \geq s_{kb} + r_{kb} - D_i - M_1(1 - \sum_{p=1}^{H_k} x_{ikbp}), \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, b \in \{1,...,B_k\} \tag{1.6}$$

$$s_{kb}, r_{kb}, u_{kb} \geq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k\} \tag{1.7}$$

$$t_i \geq 0, \quad \forall i \in \mathcal{I} \tag{1.8}$$

1.1. Equivalent KKT-based Reformulation

Define the inner minimization problem of RP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v})$ as SLP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{\boldsymbol{R}})$ under a given realization $\tilde{\boldsymbol{R}} \in \mathcal{U}(\boldsymbol{x}, \Delta)$. Let α_{ib}^k, λ_{bp}^k, β_b^k, ζ_{ib}^k, ρ_{ib}^k represent the dual variables of constraints (1.2)-(1.6) in SLP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{\boldsymbol{R}})$, respectively. We can get the dual of SLP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{\boldsymbol{R}})$ as follows.

$$\begin{aligned}
\max \; & \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} \sum_{b=1}^{B_k} \Big((\tilde{R}_i \sum_{p=1}^{H_k} x_{ikbp}) \cdot \alpha_{ib}^k + (A_i \sum_{p=1}^{H_k} x_{ikbp}) \cdot \zeta_{ib}^k + (-D_i - M_0(1 - \sum_{p=1}^{H_k} x_{ikbp})) \cdot \rho_{ib}^k \Big) \\
& + \sum_{k \in \mathcal{K}} \sum_{b=1}^{B_k} \sum_{p=1}^{H_k} \Big(\sum_{m=1}^{Z_k} U_m^k \cdot v_{kbpm} - U_k(1 - y_{kb}) \Big) \cdot \lambda_{bp}^k
\end{aligned} \tag{1.9}$$

$$s.t. \; \sum_{i \in \mathcal{I}_k} (\alpha_{ib}^k - \rho_{ib}^k) - \beta_{b+1}^k \leq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k - 1\}, \tag{1.10}$$

$$\sum_{i \in \mathcal{I}_k} (\alpha_{ib}^k - \rho_{ib}^k) \leq 0, \quad \forall k \in \mathcal{K}, b \in \{B_k\}, \tag{1.11}$$

$$\sum_{p=1}^{H_k} \lambda_{bp}^k - \beta_b^k \leq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k\}, \tag{1.12}$$

$$\sum_{i \in \mathcal{I}_k} (\zeta_{ib}^k - \rho_{ib}^k) + \beta_b^k - \beta_{b+1}^k \leq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k - 1\}, \tag{1.13}$$

$$\sum_{i \in \mathcal{I}_k} (\zeta_{ib}^k - \rho_{ib}^k) + \beta_b^k \leq 0, \quad \forall k \in \mathcal{K}, b \in \{B_k\}, \tag{1.14}$$

$$\sum_{k \in \mathcal{K}_i} \sum_{b=1}^{B_k} \rho_{ib}^k \leq C_i, \quad \forall i \in \mathcal{I}, \tag{1.15}$$

$$\alpha_{ib}^k \geq 0, \zeta_{ib}^k \geq 0, \rho_{ib}^k \geq 0, \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, b \in \{1,...,B_k\}, \tag{1.16}$$

$$\lambda_{bp}^k \geq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k\}, p = \{1,...,H_k\}, \tag{1.17}$$

$$\beta_b^k \geq 0, \quad \forall k \in \mathcal{K}, b \in \{1,...,B_k\}. \tag{1.18}$$

Let Π represent the polyhedron of the dual problem of SLP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{\boldsymbol{R}})$ and $\mathcal{P}_\Pi$ be the extreme points of Π. Note that SLP$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{\boldsymbol{R}})$ is feasible for all feasible $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v})$ and all realizations in

$\mathcal{U}(\boldsymbol{x},\boldsymbol{\Delta})$. RP$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})$ can thus be reformulated as the following maximization MINLP problem, denote as DRP$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})$.

$$\begin{aligned}
\max \ & \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}_k}(\bar{R}_{ik}+\hat{R}_{ik}\delta_i)(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\cdot\alpha_{ib}^k)+\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}_k}\sum_{b=1}^{B_k}\Big((A_i\sum_{p=1}^{H_k}x_{ikbp})\cdot\zeta_{ib}^k+(-D_i-M_0(1-\sum_{p=1}^{H_k}x_{ikbp}))\cdot\rho_{ib}^k\Big)\\
&+\sum_{k\in\mathcal{K}}\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}\Big(\sum_{m=1}^{Z_k}U_m^k\cdot v_{kbpm}-U_k(1-y_{kb})\Big)\cdot\lambda_{bp}^k\\
s.t.\ & (1.10)-(1.18)\\
& \sum_{i\in\mathcal{I}_k}(\hat{R}_{ik}\delta_i\cdot\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\leq\Delta_k,\quad\forall k\in\mathcal{K}\\
& 0\leq\delta_i\leq 1,\quad\forall i\in\mathcal{I}
\end{aligned}$$

For any given feasible $\boldsymbol{\pi}=(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\zeta},\boldsymbol{\rho})$, model DRP$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})$ reduces to

$$\max\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}_k}\hat{R}_{ik}\delta_i(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\cdot\alpha_{ib}^k)\tag{1.19}$$

$$s.t.\ \sum_{i\in\mathcal{I}_k}(\hat{R}_{ik}\delta_i\cdot\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\leq\Delta_k,\quad\forall k\in\mathcal{K}\tag{1.20}$$

$$\delta_i\leq 1,\quad\forall i\in\mathcal{I}\tag{1.21}$$

$$\delta_i\geq 0,\quad\forall i\in\mathcal{I}\tag{1.22}$$

Let η_k, $k\in\mathcal{K}$, and μ_i, $i\in\mathcal{I}$, represent the dual variables of constraints (1.20) and (1.21), respectively. The dual problem of the reduce formulation of model DRP$(\boldsymbol{x},\boldsymbol{y})$ defined in (1.19)-(1.22) can be formulated as:

$$\begin{aligned}
\min\ & \sum_{k\in\mathcal{K}}\Delta_k\eta_k+\sum_{i\in\mathcal{I}}\mu_i\\
s.t.\ & \sum_{k\in\mathcal{K}_i}\Big(\hat{R}_{ik}(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\Big)\cdot\eta_k+\mu_i\geq\sum_{k\in\mathcal{K}_i}\hat{R}_{ik}\Big(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\cdot\alpha_{ib}^k\Big),\quad\forall i\in\mathcal{I} \qquad (1.23)\\
& \eta_k\geq 0,\quad\forall k\in\mathcal{K}\\
& \mu_i\geq 0,\quad\forall i\in\mathcal{I}
\end{aligned}$$

We use $\mathcal{OU}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{\pi})$ to denote the optimal solution set of SLP$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\tilde{\boldsymbol{R}})$, which can be defined by its KKT conditions, i.e.,

$$\mathcal{OU}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{\pi})=\left\{\begin{array}{ll}
\sum_{i\in\mathcal{I}_k}(\hat{R}_{ik}\delta_i\cdot\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\leq\Delta_k, & \forall k\in\mathcal{K}\\
0\leq\delta_i\leq 1, & \forall i\in\mathcal{I}\\
\sum_{k\in\mathcal{K}_i}\Big(\hat{R}_{ik}(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\Big)\cdot\eta_k+\mu_i\geq\sum_{k\in\mathcal{K}_i}\hat{R}_{ik}\Big(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\cdot\alpha_{ib}^k\Big), & \forall i\in\mathcal{I}\\
\eta_k\geq 0, & \forall k\in\mathcal{K}\\
\mu_i\geq 0, & \forall i\in\mathcal{I}\\
\eta_k\cdot\Big(\Delta_k-\sum_{i\in\mathcal{I}_k}(\hat{R}_{ik}\delta_i\cdot\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\Big)=0, & \forall k\in\mathcal{K}\\
\mu_i\cdot(1-\delta_i)=0, & \forall i\in\mathcal{I}\\
\delta_i(\sum_{k\in\mathcal{K}_i}\Big(\hat{R}_{ik}(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\Big)\cdot\eta_k+\mu_i-\sum_{k\in\mathcal{K}_i}\hat{R}_{ik}\Big(\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\cdot\alpha_{ib}^k\Big))=0, & \forall i\in\mathcal{I}
\end{array}\right\}.\tag{1.24}$$

The first two constraints define the constraints of primal, and the next three constraints are constraints of dual problems. Others are complementary constraints, which can linearized by introducing binary variables and making use of big-M expression. As a result, the linearized form of $\mathcal{OU}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{\pi})$ is shown as below, where M and M' is a sufficiently large constant.

$$\left\{\begin{array}{ll}
\sum_{i\in\mathcal{I}_k}\varsigma_{ik}\le\Delta_k, & \forall k\in\mathcal{K}\\
\hat{R}_{ik}\delta_i-M''(1-\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp})\le\varsigma_{ik}\le\hat{R}_{ik}\delta_i+M''(1-\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}), & \forall k\in\mathcal{K},i\in\mathcal{I}_k\\
-M''\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}\le\varsigma_{ik}\le M''\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}, & \forall k\in\mathcal{K},i\in\mathcal{I}_k\\
\kappa_i\le\delta_i\le\tau_i, & \forall i\in\mathcal{I}\\
\sum_{k\in\mathcal{K}_i}\sum_{b=1}^{B_k}\psi_{ib}^k+\mu_i\ge 0, & \forall i\in\mathcal{I}\\
\hat{R}_{ik}(\eta_k-\alpha_{ib}^k)-M'(1-\sum_{p=1}^{H_k}x_{ikbp})\le\psi_{ib}^k\le\hat{R}_{ik}(\eta_k-\alpha_{ib}^k)+M'(1-\sum_{p=1}^{H_k}x_{ikbp}), & \forall i\in\mathcal{I},k\in\mathcal{K}_i,b\in\{1,2,...,B_k\}\\
-M'\sum_{p=1}^{H_k}x_{ikbp}\le\psi_{ib}^k\le M'\sum_{p=1}^{H_k}x_{ikbp}, & \forall i\in\mathcal{I},k\in\mathcal{K}_i,b\in\{1,2,...,B_k\}\\
0\le\eta_k\le M\theta_k, & \forall k\in\mathcal{K}\\
0\le\mu_i\le M\kappa_i, & \forall i\in\mathcal{I}\\
\Delta_k-\sum_{i\in\mathcal{I}_k}\varsigma_{ik}\le M(1-\theta_k), & \forall k\in\mathcal{K}\\
\sum_{k\in\mathcal{K}_i}\sum_{b=1}^{B_k}\psi_{ib}^k+\mu_i\le M(1-\tau_i), & \forall i\in\mathcal{I}\\
\theta_k\in\{0,1\}, & \forall k\in\mathcal{K}\\
\kappa_i\in\{0,1\}, & \forall i\in\mathcal{I}\\
\tau_i\in\{0,1\}, & \forall i\in\mathcal{I}
\end{array}\right\}. \tag{1.25}$$

Therefore, model RO can be equivalently written as:

$$\begin{aligned}
\text{[MP]}\quad \min\ & \sum_{k\in\mathcal{K}}\sum_{y_{kb}\in\boldsymbol{y}_k}c_k y_{kb}+\sum_{k\in\mathcal{K}}\sum_{\sigma_{kb}\in\boldsymbol{\sigma}_k}f_k\sigma_{kb}+\phi\\
s.t.\ & (\boldsymbol{y}_k,\boldsymbol{\sigma}_k,\boldsymbol{v}_k)\in\mathcal{Q}(\boldsymbol{x},k,B_k,H_k),\quad\forall k\in\mathcal{K}\\
& \boldsymbol{x}\in\mathcal{X}\\
& \phi\ge\sum_{i\in\mathcal{I}}C_i t_i^{(\pi)},\quad\forall\boldsymbol{\pi}\in\mathcal{P}_\Pi,\\
& (\boldsymbol{s}^{(\boldsymbol{\pi})},\boldsymbol{r}^{(\boldsymbol{\pi})},\boldsymbol{u}^{(\boldsymbol{\pi})},\boldsymbol{t}^{(\boldsymbol{\pi})})\in\mathcal{H}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\tilde{R}_{\boldsymbol{\delta_\pi}}),\quad\forall\boldsymbol{\pi}\in\mathcal{P}_\Pi,\\
& \tilde{R}_{\boldsymbol{\delta_\pi}}=\sum_{k\in\mathcal{K}}[(\bar{R}_{ik}+\hat{R}_{ik}\delta_i^{(\boldsymbol{\pi})})\cdot\sum_{b=1}^{B_k}\sum_{p=1}^{H_k}x_{ikbp}],\quad\forall\boldsymbol{\pi}\in\mathcal{P}_\Pi,\\
& (\boldsymbol{\delta}^{(\boldsymbol{\pi})},\boldsymbol{\eta}^{(\boldsymbol{\pi})},\boldsymbol{\mu}^{(\boldsymbol{\pi})},\boldsymbol{\theta}^{(\boldsymbol{\pi})},\boldsymbol{\kappa}^{(\boldsymbol{\pi})},\boldsymbol{\tau}^{(\boldsymbol{\pi})})\in\mathcal{OU}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{\pi}),\quad\forall\boldsymbol{\pi}\in\mathcal{P}_\Pi,
\end{aligned}\tag{1.26}$$

1.2. Algorithm Details

The parameterized C&CG algorithm is summarized in Algorithm 1. Denote $\hat{\mathcal{P}}_\Pi$ as the subset of $\mathcal{P}_\Pi$ and set the initial $\hat{\mathcal{P}}_\Pi$ as an empty set. In each iteration n, the parameterized C&CG algorithm first solve the master problem defined in (1.27) based on the subset $\hat{\mathcal{P}}_\Pi\subseteq\mathcal{P}_\Pi$. The optimal objective value of MP provides a lower bound on the optimal objective value of model RO. Let $\boldsymbol{x}^*$ indicate the optimal service plan obtained for the master problem. For the solution $\boldsymbol{x}^*$, the parameterized C&CG algorithm solve the subproblem defined in (1.28) to obtain the worst-case second-stage cost and the associate solution $\boldsymbol{\pi}^*=(\boldsymbol{\alpha}^*,\boldsymbol{\lambda}^*,\boldsymbol{\beta}^*,\boldsymbol{\zeta}^*,\boldsymbol{\rho}^*)$. The upper bound of the algorithm is updated accordingly. If the lower bound equals the upper bound, then model RO is solved to optimum, and our C&CG algorithm terminates with an optimal solution indicated by $(\boldsymbol{x}^*,\boldsymbol{y}^*,\boldsymbol{\sigma}^*,\boldsymbol{v}^*)$. Otherwise, it appends the identified $\boldsymbol{\pi}^*$ to the subset $\hat{\mathcal{P}}_\Pi$. We note that the dynamic parameter adjusting strategy and variable reduction strategies proposed in this paper can be direclty applied to the parameterized C&CG algorithm.

Algorithm 1: Parameterized C&CG Algorithm

1. Set initial $\hat{\mathcal{P}}_\Pi = \emptyset$, UB=$+\infty$ and LB=$-\infty$.

2. Solve the following master problem to obtain the solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*)$ and update LB

$$\begin{aligned}
\min \ & \sum_{k\in\mathcal{K}} \sum_{y_{kb}\in\boldsymbol{y}_k} c_k y_{kb} + \sum_{k\in\mathcal{K}} \sum_{\sigma_{kb}\in\boldsymbol{\sigma}_k} f_k \sigma_{kb} + \phi \\
s.t. \ & (\boldsymbol{y}_k, \boldsymbol{\sigma}_k, \boldsymbol{v}_k) \in \mathcal{Q}(\boldsymbol{x}, k, B_k, H_k), \quad \forall k \in \mathcal{K} \\
& \boldsymbol{x} \in \mathcal{X} \\
& \phi \geq \sum_{i\in\mathcal{I}} C_i t_i^{(\pi)}, \quad \forall \boldsymbol{\pi} \in \mathcal{P}_\Pi, \\
& (\boldsymbol{s}^{(\boldsymbol{\pi})}, \boldsymbol{r}^{(\boldsymbol{\pi})}, \boldsymbol{u}^{(\boldsymbol{\pi})}, \boldsymbol{t}^{(\boldsymbol{\pi})}) \in \mathcal{H}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \tilde{R}_{\boldsymbol{\delta}_{\boldsymbol{\pi}}}), \quad \forall \boldsymbol{\pi} \in \hat{\mathcal{P}}_\Pi, \\
& (\boldsymbol{\delta}_{\boldsymbol{\pi}}, \boldsymbol{\eta}_{\boldsymbol{\pi}}, \boldsymbol{\mu}_{\boldsymbol{\pi}}, \boldsymbol{\theta}_{\boldsymbol{\pi}}, \boldsymbol{\kappa}_{\boldsymbol{\pi}}, \boldsymbol{\tau}_{\boldsymbol{\pi}}) \in \mathcal{OU}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\pi}), \quad \forall \boldsymbol{\pi} \in \hat{\mathcal{P}}_\Pi.
\end{aligned} \tag{1.27}$$

3. Solve the following subproblem to obtain the optimal $\boldsymbol{\delta}^*$ and $\boldsymbol{\pi}^* = (\boldsymbol{\alpha}^*, \boldsymbol{\lambda}^*, \boldsymbol{\beta}^*, \boldsymbol{\zeta}^*, \boldsymbol{\rho}^*)$ and update UB

$$\begin{aligned}
\max \ & \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}_k} \sum_{b=1}^{B_k} \Big((\bar{R}_{ik} \sum_{p=1}^{H_k} x_{ikbp}) \cdot \alpha_{ib}^k + (A_i \sum_{p=1}^{H_k} x_{ikbp}) \cdot \zeta_{ib}^k + (-D_i - M_0(1 - \sum_{p=1}^{H_k} x_{ikbp})) \cdot \rho_{ib}^k \Big) \\
& + \sum_{k\in\mathcal{K}} \sum_{b=1}^{B_k} \sum_{p=1}^{H_k} \Big(\sum_{m=1}^{Z_k} U_m^k \cdot v_{kbpm} - U_k(1 - y_{kb}) \Big) \cdot \lambda_{bp}^k + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}_k} \varphi_i^k \\
s.t. \ & (1.10) - (1.18) \\
& \ell \cdot \Big(\sum_{b=1}^{B_k(\boldsymbol{x})} \sum_{p=1}^{H_k} (x_{ikbp} \cdot \alpha_{ib}^k) \Big) - M(1 - \gamma_{i\ell}^k) \leq \varphi_i^k \leq \ell \cdot \Big(\sum_{b=1}^{B_k(\boldsymbol{x})} \sum_{p=1}^{H_k} (x_{ikbp} \cdot \alpha_{ib}^k) \Big) + M(1 - \gamma_{i\ell}^k), \\
& \qquad \forall k \in \mathcal{K}, i \in \mathcal{I}_k(\boldsymbol{x}), \ell \in \{0, ..., \min\{\hat{R}_{ik}, \Delta_k\}\}, \\
& \sum_{\ell=0}^{\min\{\hat{R}_{ik}, \Delta_k\}} \gamma_{i\ell}^k = 1, \quad \forall \sum_{k\in\mathcal{K}}, i \in \mathcal{I}_k(\boldsymbol{x}), \\
& \sum_{i\in\mathcal{I}_k(\boldsymbol{x})} \sum_{\ell=0}^{\min\{\hat{R}_{ik}, \Delta_k\}} \ell \cdot \gamma_{i\ell}^k \leq \Delta_k, \quad \forall k \in \mathcal{K}, \\
& \gamma_{i\ell}^k \in \{0, 1\}, \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k(\boldsymbol{x}), \ell \in \{0, ..., \min\{\hat{R}_{ik}, \Delta_k\}\}.
\end{aligned} \tag{1.28}$$

4. If UB=LB, algorithm stops with optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*)$. Otherwise update $\hat{\mathcal{P}}_\Pi = \hat{\mathcal{P}}_\Pi \cup \{\boldsymbol{\pi}^*\}$ and go to the next iteration.

2. A First-Come First-Served Method

The shipyard operators currently mainly utilize a First-Come First-Served (FCFS) method for dry dock scheduling. The FCFS method assigns ships one by one to the dry docks according to an ascending order of their arrival times. As the FCFS method used by the shipyard operators incorporates various specific operation considerations, following Jia et al. (2024), here we consider an simplified FCFS method with assuming that the shipyard operators can exactly estimate the actual arrival times when making decisions. In this appendix, we provide the pseudo-code of the first-come first-served method. In addition to the notations used in model RO, we introduce the following notations:

$\mathcal{I}' = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$: List of ships sorted in ascending order of A_i, where i_μ $(\mu = 1, \ldots, |\mathcal{I}|)$ is the μth ship.

$\mathcal{I}_{kb}$: Set of ships assigned to the bth batch of dry dock k.

$\tilde{\mathcal{I}}_{kb}$: Tentative value of $\mathcal{I}_{kb}$.

N_k: Number of batches assigned to dry dock k.

$\chi_k(i) := 1$ if ship i is served in parallel with other ships in dry dock k; 0 otherwise.

$z(i)$: Block layout used by ship i.

$L_{z(i)}$: Length of block layout used by ship i.

$y_{kb} := 1$ if the batch b of dry dock k need rearrange the block; 0 otherwise.

R_{kb}: Service time of the bth batch in dry dock k, $R_{kb} = \max_{i \in \mathcal{I}_{kb}} \{\bar{R}_{ki}\}$ and $\bar{R}_{ki}$ is the normal service time of ship i in dock k.

$\tilde{R}_{kb}$: Tentative value of R_{kb}.

$u(i)$: Setup time of the block layout used by ship i.

U_{kb}: Setup time of all block layouts used in the bth batch of dry dock k.

$\tilde{U}_{kb}$: Tentative value of U_{kb}.

s_{kb}: Service start time of the bth batch of dry dock k.

$\tilde{s}_{kb}$: Tentative value of s_{kb}.

$\mathcal{Z}_{kb}$: Set of block layouts used by ships in the bth batch of dry dock k.

F_{Total}: Total cost, including the setup cost, operational cost of each batch and the tardiness cost of each ship.

F_{kb}: Total cost incurred by current ships in the bth batch of dry dock k.

$F_{kb}(i)$: Total cost of the bth batch of dry dock k if ship i is added to this batch.

$\psi_k(i)$: Increment in the total cost resulted from assigning ship i to dry dock k.

The pseudo-code of the first-come first-served method is provided as follows:

In algorithm 2, step 1 initializes the dry dock schedules by assigning a dummy batch to each dry dock. Steps 2–35 constitute the main loop that iteratively assigns one ship in the sorted ship list to a dry dock and compute the total cost. Within the main loop, steps 3–22 form an inner loop that checks whether ship i_μ can be served in each dry dock k, and if it is assigned to dry dock k, whether it should be served independently (served in batch $b+1$) or be served in parallel with other ships in the last batch (served in batch b). Specifically, steps 4–11 determine the increment in the total cost if ship i_μ is served independently in dry dock k, while steps 12–22 determine the increment in the total cost if ship i_μ is served in parallel with other ships in the last batch of dry dock k. Step 16 is used to verify whether the layout of the current batch is compatible with the previous batch. If not compatible, additional setup costs will be incurred. If ship i_μ serves in batch $b+1$, the increment in total cost is $F_{k,b+1}(i_\mu)$, and if ship i continues to be served in parallel in batch b, the increment in total cost is $F_{kb}(i_\mu) - F_{kb}$. By comparing the cost increments of these two cases, the values of $\chi_k(i_\mu)$ and $\psi_k(i_\mu)$ can be determined. After the inner loop, step 23 selects the optimal dry dock k^* for ship i_μ and steps 24-34 update the work schedule of dry dock k^*. Specifically, the best dry dock k^* for ship i_μ is the one that is compatible with the ship and that results the minimum increment in the total cost. If ship i_μ is served independently in dry dock k^* (i.e., $\chi_k(i_\mu) = 0$), then the batch count of dry dock

Algorithm 2: First-come first-served method

Set $F_{\text{Total}} \leftarrow 0$. Set $N_k \leftarrow 0, s_{k0} \leftarrow 0, R_{k0} \leftarrow 0$ and $\mathcal{Z}_{k0} \leftarrow \emptyset$ for each $k \in \mathcal{K}$.
for $\mu = 1, \ldots, |\mathcal{I}|$ **do**
  for $k \in \mathcal{K}$ such that $i_\mu \in \mathcal{I}_k$ and $L_{z(i_\mu)} \leq \bar{L}_k$ **do**
    Set $b \leftarrow N_k$ and $\tilde{\mathcal{I}}_{k,b+1} \leftarrow \{i_\mu\}$.
    if $z(i_\mu) \in \mathcal{Z}_{kb}$ **then**
      Set $y_{k,b+1} \leftarrow 0$ and $\tilde{U}_{k,b+1} \leftarrow 0$.
    else
      Set $y_{k,b+1} \leftarrow 1$ and $\tilde{U}_{k,b+1} \leftarrow u(i_\mu)$.
    Set $\tilde{s}_{k,b+1} \leftarrow \max\{A_{i_\mu}, s_{kb} + R_{kb} + \tilde{U}_{k,b+1}\}$ and $\tilde{R}_{k,b+1} \leftarrow \bar{R}_{k,i_\mu}$.
    Set $F_{k,b+1}(i_\mu) \leftarrow c_k y_{k,b+1} + f_k + C_{i_\mu} \max\{0, \tilde{s}_{k,b+1} + \tilde{R}_{k,b+1} - D_{i_\mu}\}$.
    Set $\chi_k(i_\mu) \leftarrow 0$, and $\psi_k(i_\mu) \leftarrow F_{k,b+1}(i_\mu)$.
    if $b > 0$ and $L_{z(i_\mu)} + \sum_{i \in \mathcal{I}_{kb}} L_{z(i)} \leq \bar{L}_k$ **then**
      Set $F_{kb} \leftarrow c_k y_{kb} + f_k + \sum_{i \in \mathcal{I}_{kb}} C_i \max\{0, s_{kb} + R_{kb} - D_i\}$.
      Set $\tilde{\mathcal{I}}_{kb} \leftarrow \mathcal{I}_{kb} \cup \{i_\mu\}$ and $\tilde{U}_{kb} \leftarrow \max_{i \in \tilde{\mathcal{I}}_{kb}}\{u(i)\}$.
      Set $\tilde{s}_{kb} \leftarrow \max\{A_{i_\mu}, s_{k,b-1} + R_{k,b-1} + \tilde{U}_{kb}\}$, and $\tilde{R}_{kb} \leftarrow \max_{i \in \tilde{\mathcal{I}}_{kb}}\{\bar{R}_{ki}\}$.
      if $\{z(i) \mid i \in \tilde{\mathcal{I}}_{kb}\} \subseteq \mathcal{Z}_{k,b-1}$ **then**
        Set $y_{kb} \leftarrow 0$.
      else
        Set $y_{kb} \leftarrow 1$.
      Set $F_{kb}(i_\mu) \leftarrow c_k y_{kb} + f_k + \sum_{i \in \tilde{\mathcal{I}}_{kb}} C_i \max\{0, \tilde{s}_{kb} + \tilde{R}_{kb} - D_i\}$.
      if $F_{kb}(i_\mu) - F_{kb} < F_{k,b+1}(i_\mu)$ **then**
        Set $\chi_k(i_\mu) \leftarrow 1$ and $\psi_k(i_\mu) \leftarrow F_{kb}(i_\mu) - F_{kb}$.
  Set $k^* \leftarrow \arg\min_{k \in \mathcal{K}: i_\mu \in \mathcal{I}_k}\{\psi_k(i_\mu)\}$ and $b \leftarrow N_{k^*}$.
  if $\chi_{k^*}(i_\mu) = 0$ **then**
    Set $N_{k^*} \leftarrow N_{k^*} + 1$ and $b \leftarrow N_{k^*}$.
    Set $\mathcal{I}_{k^*b} \leftarrow \{i_\mu\}$, $R_{k^*b} \leftarrow \bar{R}_{k,i_\mu}$, $U_{k^*b} \leftarrow u(i_\mu)$ and $\mathcal{Z}_{k^*b} \leftarrow \{z(i_\mu)\}$.
  else
    Set $\mathcal{I}_{k^*b} \leftarrow \mathcal{I}_{k^*b} \cup \{i_\mu\}, R_{k^*b} \leftarrow \max_{i \in \mathcal{I}_{k^*b}}\{\bar{R}_{ki}\}$ and $U_{k^*b} \leftarrow \max_{i \in \mathcal{I}_{k^*b}}\{u(i)\}$.
    Set $\mathcal{Z}_{k^*b} \leftarrow \{z(i_\mu) \mid i \in \mathcal{I}_{k^*b}\}$.
  if $\mathcal{Z}_{k^*b} \subseteq \mathcal{Z}_{k^*,b-1}$ **then**
    Set $y_{k^*b} \leftarrow 0$.
  else
    Set $y_{k^*b} \leftarrow 1$.
  Set $s_{k^*b} \leftarrow \max\{A_{i_\mu}, s_{k^*,b-1} + R_{k^*,b-1} + U_{k^*b}\}$.
  Set $F_{\text{Total}} \leftarrow F_{\text{Total}} + \psi_{k^*}(i_\mu)$.

k^* is increased by 1. Otherwise, the batch count for dry dock k^* remains unchanged. At the same time, the service time, setup time, layout set, and service start time of the selected batch need to be updated based on the ships within the batch. After the assignment of ship i_μ, the total cost is increased by $\psi_{k^*}(i_\mu)$.

3. Two Benchmark Models for Best Achievable Outcomes

We also compare model RO with two benchmark models: A stochastic programming model and a min-max optimization model. Here, we provide the details of the two benchmark models.

Given a set Π of realizations of service times which are assumed to be uniformly distributed, we can formulate a stochastic programming as following MILP, whose objective function seeks to minimize the expected total cost over all realizations in Π.

$$
\begin{aligned}
\text{[SP]}\quad \min\ & \sum_{\tilde{\boldsymbol{R}}\in\Pi}\frac{1}{|\Pi|}\cdot\Big(\sum_{k\in\mathcal{K}}\sum_{y_{kb}\in\boldsymbol{y}_k}c_k y_{kb}+\sum_{k\in\mathcal{K}}\sum_{\sigma_{kb}\in\boldsymbol{\sigma}_k}f_k\sigma_{kb}+\sum_{i\in\mathcal{I}}C_i t_i^{(\tilde{\boldsymbol{R}})}\Big)\\
s.t.\ & \boldsymbol{x}\in\mathcal{X}\\
& (\boldsymbol{y}_k,\boldsymbol{\sigma}_k,\boldsymbol{v}_k)\in\mathcal{Q}(\boldsymbol{x},k,B_k,H_k),\quad \forall k\in\mathcal{K},\\
& r_{kb}^{(\tilde{\boldsymbol{R}})}\ge\tilde{R}_{ik}^{(\omega)}\sum_{p=1}^{H_k}x_{ikbp},\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& u_{kb}^{(\tilde{\boldsymbol{R}})}\ge\sum_{m=1}^{M}U_m^k\cdot v_{kbpm}-U_k(1-y_{kb}),\quad \forall k\in\mathcal{K},b\in\{1,...,\hat{B}_k\},p=\{1,...,\hat{H}_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& s_{kb}^{(\tilde{\boldsymbol{R}})}\ge s_{k,b-1}^{(\tilde{\boldsymbol{R}})}+r_{k,b-1}^{(\tilde{\boldsymbol{R}})}+u_{kb}^{(\tilde{\boldsymbol{R}})},\quad \forall k\in\mathcal{K},b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& s_{kb}^{(\tilde{\boldsymbol{R}})}\ge A_i\sum_{p=1}^{H_k}x_{ikbp},\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& t_i^{(\tilde{\boldsymbol{R}})}\ge s_{kb}^{(\tilde{\boldsymbol{R}})}+r_{kb}^{(\tilde{\boldsymbol{R}})}-D_i-M_3(1-\sum_{p=1}^{H_k}x_{ikbp}),\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& s_{kb}^{(\tilde{\boldsymbol{R}})},r_{kb}^{(\tilde{\boldsymbol{R}})},u_{kb}^{(\tilde{\boldsymbol{R}})}\ge 0,\quad \forall k\in\mathcal{K},b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& t_i^{(\tilde{\boldsymbol{R}})}\ge 0,\quad \forall i\in\mathcal{I},\tilde{\boldsymbol{R}}\in\Pi.
\end{aligned}
$$

Similarly, given Π, we can formulate a min-max optimization model as the following MILP, which aims to minimize the worst-case total cost over all possible realizations in Π.

$$
\begin{aligned}
\text{[MM]}\quad \min\ & \sum_{k\in\mathcal{K}}\sum_{y_{kb}\in\boldsymbol{y}_k}c_k y_{kb}+\sum_{k\in\mathcal{K}}\sum_{\sigma_{kb}\in\boldsymbol{\sigma}_k}f_k\sigma_{kb}+\phi\\
s.t.\ & \boldsymbol{x}\in\mathcal{X}\\
& (\boldsymbol{y}_k,\boldsymbol{\sigma}_k,\boldsymbol{v}_k)\in\mathcal{Q}(\boldsymbol{x},k,B_k,H_k),\quad \forall k\in\mathcal{K},\\
& \phi\ge\sum_{i\in\mathcal{I}}C_i t_i^{(\tilde{\boldsymbol{R}})},\quad \forall\tilde{\boldsymbol{R}}\in\Pi,\\
& r_{kb}^{(\tilde{\boldsymbol{R}})}\ge\tilde{R}_{ik}^{(\omega)}\sum_{p=1}^{H_k}x_{ikbp},\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& u_{kb}^{(\tilde{\boldsymbol{R}})}\ge\sum_{m=1}^{M}U_m^k\cdot v_{kbpm}-U_k(1-y_{kb}),\quad \forall k\in\mathcal{K},b\in\{1,...,\hat{B}_k\},p=\{1,...,\hat{H}_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& s_{kb}^{(\tilde{\boldsymbol{R}})}\ge s_{k,b-1}^{(\tilde{\boldsymbol{R}})}+r_{k,b-1}^{(\tilde{\boldsymbol{R}})}+u_{kb}^{(\tilde{\boldsymbol{R}})},\quad \forall k\in\mathcal{K},b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& s_{kb}^{(\tilde{\boldsymbol{R}})}\ge A_i\sum_{p=1}^{H_k}x_{ikbp},\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,\\
& t_i^{(\tilde{\boldsymbol{R}})}\ge s_{kb}^{(\tilde{\boldsymbol{R}})}+r_{kb}^{(\tilde{\boldsymbol{R}})}-D_i-M_3(1-\sum_{p=1}^{H_k}x_{ikbp}),\quad \forall k\in\mathcal{K},i\in\mathcal{I}_k,b\in\{1,...,B_k\},\tilde{\boldsymbol{R}}\in\Pi,
\end{aligned}
$$

$$s_{kb}^{(\tilde{\boldsymbol{R}})}, r_{kb}^{(\tilde{\boldsymbol{R}})}, u_{kb}^{(\tilde{\boldsymbol{R}})} \geq 0, \quad \forall k \in \mathcal{K}, b \in \{1, ..., B_k\}, \tilde{\boldsymbol{R}} \in \Pi,$$

$$t_i^{(\tilde{\boldsymbol{R}})} \geq 0, \quad \forall i \in \mathcal{I}, \tilde{\boldsymbol{R}} \in \Pi.$$

References

Jia, S., Li, C.L., Meng, Q., 2024. The dry dock scheduling problem. Transportation Research Part B: Methodological 181, 102893.

Zeng, B., Wang, W., 2022. Two-stage robust optimization with decision dependent uncertainty. arXiv preprint arXiv:2203.16484 .